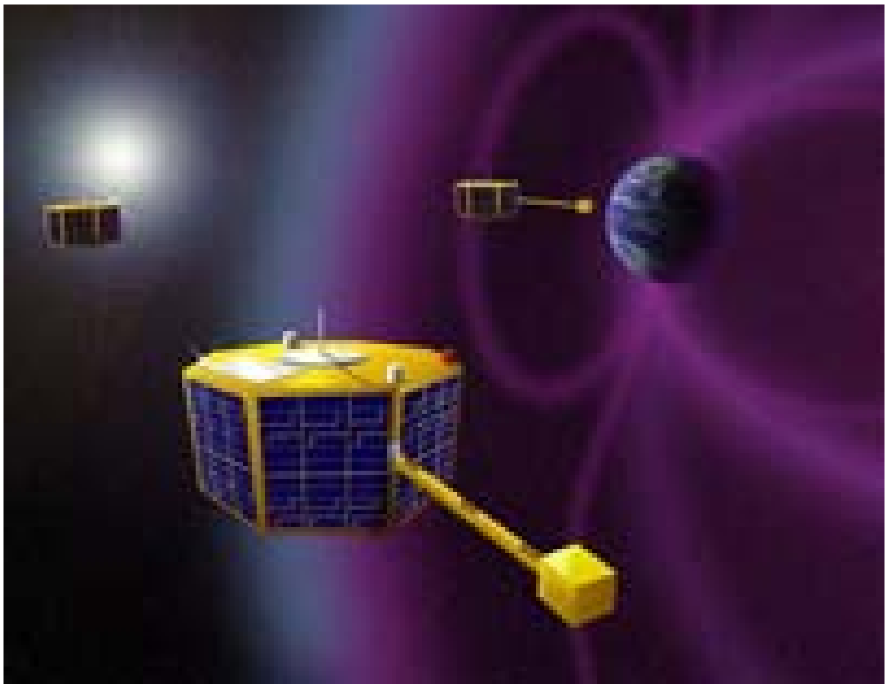
GSFC MQP - A'02

Spacecraft Model Development

10/08/02



<u>Submitted By:</u>
Paul Blanchard
(pblanch@wpi.edu)

Sarah Casey
(sjcfifer@wpi.edu)

Allison Rhorer
(arhorer@wpi.edu)

Goddard Space Flight Center
Building #23, Room C322
Phone: 301-286-9742
<u>GSFC Mentor:</u>
Irene Bibyk
(ibibyk@pop500.GSFC.NASA.GOV)
Phone: (301) 286-6213
<u>WPI Advisor:</u>
Professor Fred J. Looft
(fjlooft@ece.wpi.edu)

A Project Report submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements
for the Degree of Bachelor of Science.

<u>TABLE OF CONTENTS</u>

TABLE OF FIGURES

Abstract

Matlab and Simulink were used to create a behavioral model of the Space Technology 5 (ST-5) satellites, which are scheduled for launch after 2003. The model focused on the health status of the satellite power system and data recorder, and on the satellite's ability to transmit at a given time. The model will serve as a testing and updating tool for the Goddard Space Flight Center. The project idea was generated from a similar project performed by the United States Air Force Academy in which requirements and behavioral models were created and implemented for the FalconSat small satellite program.

1. Project Introduction

1.1 Introduction

The National Aeronautics and Space Administration (NASA) is currently exploring and testing highly advanced technologies as part of the New Millennium Program (NMP). The NMP was created as a result of NASA's high expectations for its space program during this millennium, including the desire to speed up the pace of space exploration. One of the components of the NMP is Space Technology 5 (ST-5), a technology-driven mission that will involve three (3) micro-satellites launched together and flown in a string-of-pearls formation. ST-5 will serve to validate, in flight, several new technologies as part of the NMP and is expected to launch sometime after 2003. Due to the technological focus of this mission, NASA desires the ability to test different flight scenarios in a dynamic environment, in order to determine the behavior of the spacecraft components and also to verify that the satellite design will sufficiently meet the operational requirements.

The United States Air Force Academy (USAFA) has a satellite program, FalconSat, intended for students with little or no design experience and limited funds for the design and testing processes. A USAFA cadet, Stuart Stanton, involved in the FalconSat program developed and successfully implemented a method of satellite simulation using block diagrams and commercial simulation software, in order to reduce the cost and time requirements of the development program. The modeling was used to ensure that the satellite design sufficiently met the mission and operational requirements. For the FalconSat program, the USAFA cadets created two (2) models: a requirements model and a behavioral model. The requirements model showed static relationships between subsystems and overall performance feedback for the entire system; this model was developed using Microsoft Excel. The behavioral model was used to simulate dynamic interactions between subsystems in a simulated operational environment; this model was developed using Simulink. The behavioral model was also used to provide feedback on orbital mechanics, sunlight/eclipse cycles, and payload duty cycles. The models developed by the USAFA cadets were flexible and evolvable, providing a simulation tool for design purposes and mission planning.

NASA made plans to create a model, similar to the behavioral model designed by the USAFA, of ST-5. The model was created using block diagrams and commercial software, primarily Simulink. The ST-5 model will be used to verify power supply and energy consumption as a function of orbit and operational scenarios, in order to ensure that the satellite design is sufficiently robust to allow the satellites to operate at all points in the projected orbit. The satellite model will also be used to verify that the data recorder will not overflow during operational scenarios and that the satellite can transmit at scheduled transmission times. Behavioral modeling will lower program cost by providing an integrated tool to reduce design errors and test operational capabilities prior to implementation, and will provide a means for meeting some of the goals of the NMP and ST-5 programs, lowering system costs and reducing spacecraft mission development time.

1.2 Project Statement

The overall purpose of this project was to create an accurate and flexible behavioral model of one of the ST-5 micro-satellites using Simulink. The model was designed to verify requirements before implementation in the satellite. It was envisioned that the model would provide a way to monitor the status of some of the satellites' subsystems, mainly the power, the communications, and the data recorder, and also contain an orbit propagator to generate position and velocity information. The behavioral model was therefore determined to contain four sub-models: an orbit propagator model, a power subsystem model, a data recorder model, and a communication subsystem model. The behavioral model would also serve as an operational tool for NASA, aiding in the identification of any errors or potential failures the ST-5 mission may have.

Power margin is a major concern in this mission due to the small size of the satellites and the power subsystem. One of the main applications of this model was to ensure that the satellite's power margin was sufficient at all points in the orbit. Since the amount of energy stored in a battery is limited by its size, the battery may have a problem maintaining power to the satellite during an eclipse period. A pre-existing power subsystem model was the foundation for the completed behavioral model. In the future, the developed behavioral model will also be used for verification of any changed elements of the spacecraft launch.

The first step toward model development was to integrate a pre-existing orbital propagation model into the power model. The orbital propagation model used initial position and velocity vectors to determine the orbit of the satellite. Our first goal was to use the orbital information, in conjunction with the computed position of the sun, to determine the satellites' eclipse periods.

While the satellite is in eclipse, all subsystems except the Command and Data Handling (C&DH), RF transponder in receive mode, Miniature Spinning Sun Sensor (MSSS), and possibly the magnetometer, will be inactive. An onboard lithium ion (Li-Ion) battery will power the satellite during this time. While the satellite is illuminated, the solar panels will generate current used to power the spacecraft and to recharge the battery.

The power subsystem model, developed by Scott Starin at Goddard Space Flight Center (GSFC), was updated in order to make it more useful. One problem with the existing power subsystem model was that the solar array (SA) power output was modeled as a constant set equal to the expected average solar array output; our goal was to create a more dynamic model of the solar cells. The dynamic model made it possible to determine the actual power that the SA was generating. The new dynamic model also allowed the user would then be able to simulate the failure of one of the SA panels, which may occur during flight.

The existing orbit propagator model was updated to output the satellite Earth-Centered Inertial (ECI) coordinates and whether the satellite is illuminated or eclipsed. The outputs of the orbit propagator are used in conjunction with the Communications Model, which determines the link margin, Doppler shift, and whether it is possible to transmit. If a transmission is scheduled and if it is possible to transmit (sufficient link margin, and grounds station sight and availability), then the high power amplifier (HPA) is activated and transmission occurs.

A data recorder model was developed to track the rate of data collection, amount of memory available, and the downlink rate of the transponder. The data recorder model will allow us to determine whether or not data overflow occurred and whether there was a sufficient downlink window to transmit all the data to the ground station. The data recorder model also allowed us to predict the amount of data collected between downlink periods and determine whether or not any data was lost.

A communications model was created to calculate the link margin, Doppler shift, and the ground station visibility, at times when transmission is desired. The communications model allowed us to determine which ground station(s) could be transmitted to at any time that transmission was desired. The ability to transmit to a ground station requires that the satellite be visible to the ground station antenna and that there be sufficient link margin for the transmission to succeed.

Once the subsystem models discussed above were created and integrated, we began expanding on the models. One priority was to create and test the effect of a more realistic model of the voltage regulator that incorporates pulse-width modulation and compare the results to the pre-existing regulator model.

1.3 Summary

A model was created for NASA's ST-5 spacecraft that was based on a model created by USAFA cadets for the FalconSat small satellite program. The ST-5 model is comprised of 4 sub-models: an orbit propagator model that is run separately from the other sub-models; a power subsystem model that monitors the state of the power subsystem and the power margin; a data recorder model that models the state of the data recorder and tracks the amount of data in the recorder; and a communications model which calculates the link margin, Doppler shift, and the ground station visibility for those times when the satellite will be transmitting. The model was designed to be used as a mission-planning tool, to ensure that desired operations can be executed successfully, and also as a design verification tool, to ensure that the satellite operates as expected.

2. Background

2.1 Introduction

In order to fully understand the goals and objectives of the project, a brief overview of satellites in general and the ST-5 satellites specifically will be provided. This overview includes a history of satellites, the background of the ST-5 mission, the basics of spacecraft flight, the specifics of the ST-5 satellite subsystems that are relevant to the behavioral model, and an explanation of the simulation software used to create the model. This background aided in expanding our general knowledge of space exploration, satellite technology, ST-5 specific goals, and the tools that were used to generate the final behavioral model.

2.2 History and Development of Satellites

"History changed on October 4, 1957, when the Soviet Union successfully launched Sputnik I. The world's first artificial satellite was about the size of a basketball, weighed only 183 pounds, and took approximately 98 minutes to orbit the Earth on an elliptical path. That single launch opened the doors for new political, military, technological, and scientific developments." -Roger D. Launius, NASA Chief Historian

Since that time, thousands of satellites have been successfully launched by various countries and for a variety of reasons. Applications of satellites include communications, weather prediction, navigation, conducting experiments in zero gravity, and observing the stars. Governments are not the only entities who are launching satellites. Many businesses, universities and other organizations around the world also have an interest in launching and managing their own satellites. Various organizations even offer individuals the chance to launch their own satellites.

With all the different interests and uses, the satellites themselves vary a great deal in their capabilities and mission objectives. The primary tradeoff between different satellites is size verses functionality and dependability. Developing and launching any satellite is a very expensive process.

Up until the 1990's, the satellites that were being produced were increasing in size. One cause of this was the trend of increasingly ambitious missions and goals, which required larger satellites. Another reason was that the payload capability of launch vehicles was making it increasingly easier to launch larger satellites. While these large satellites were designed to be dependable, they were also very expensive. After the complete or partial failure of several high-profile, multi-billion dollar, science spacecraft missions - including the loss of Mars Observer, the flaw in the Hubble Space Telescope's primary mirror, and the failed deployment of Galileo's high gain antenna - a new way of designing satellites emerged.

2.3 SmallSats, FalconSat, and the Nanosat Constellation Trailblazer

Highly functional, light-weight mini-satellites are categorized according to their weight, with micro-satellites weighing between 20 and 200 lbs, nano-satellites weighing between 2 and 20 lbs, and pico-satellites weighing less than 2 lbs. Nano- and pico-satellites are envisioned as the future of satellite technology, with swarms launched to perform a range of tasks. Some of these tasks are currently performed by single, expensive, traditional satellites.

The FalconSat Program at the United States Air Force Academy (USAFA) is an example of a small satellite program. FalconSat is a continuation and redefinition of the USAFA's SmallSat program, begun in 1991. Several satellite prototypes were launched by the USAFA. The first of the prototypes to fly was USAFASAT-B, which successfully flew more than 100K ft above the earth's surface in May of 1995, launched by a balloon (Stanton, 1999).

Falcon Gold was launched in October of 1997, and transmitted data until November, 1997. The launch took place from Cape Canaveral Air Station, FL, and was an 'unqualified success' (Stanton, 1999). In January of 2000, the Joint Academy Weber State Satellite (JAWSAT) was launched from Vandenberg AFB, California. JAWSAT was collaboration between students at Weber State University and cadets at the USAFA.

Upon reaching orbit, JAWSAT deployed four smaller satellites (ASUSAT, OPAL, OCSE, and FalconSat1). FalconSat2 is currently being developed by cadets at the USAFA (Stanton, 1999).

Small satellites are not just for students. After focusing on larger, more expensive spacecraft for many years, NASA is beginning to return its focus now to smaller, cheaper satellites. The buzzwords of 'faster, cheaper, better' are becoming increasingly important as both budgets and new technology become smaller (Sarsfield, 1998). One of the programs adopting this philosophy is Space Technology 5 (ST-5) with the Nanosat Constellation Trailblazer family of satellites.

The Nanosat Constellation Trailblazer (NCT) satellites are small, octagonal satellites. They are approximately 16 inches across, 8 inches high, and weigh only 44 lbs each. The satellites will be launched as a secondary payload and will contain experimental technology that will be tested in space. Such testing is necessary to ensure that the components perform as expected and is the cornerstone of the New Millennium Project and the ST-5 mission (About ST-5, 2002).

2.4 Space Technology 5 and the New Millennium Project

The New Millennium Program (NMP) was created to support the high expectations that NASA has for its space program in this millennium. NASA's vision for this millennium is to speed up the rate of space exploration through the development of highly advanced satellite technologies. NMP is different from NASA's other advanced technology projects in that it will test the new technologies during space flight, rather than in ground labs. The unique space-based testing is a result of the nature of several of the technologies and concepts that NMP addresses, including solar electric (ion) propulsion, or, in the case of ST-5, multiple spacecraft flying in formation. Other goals that NMP has are to:

- Reduce the size of spacecraft, in order to decrease cost.
- Create "intelligent" spacecraft; in order to minimize the amount of handling required by ground personnel.

- Enable significantly improved technical capabilities in future missions.

Along with ST-5, there are four other projects that are part of the New Millennium Program. All five projects share the three supporting goals listed above, but target different technology development areas.

The ST-5 program will involve the simultaneous launch of three (3) fully functional, autonomous micro-satellites, to be flown in formation as if they were a single satellite. The satellites will spend three (3) months in a geosynchronous transfer orbit. The satellites will be powered by an array of solar cells and a rechargeable Lithium Ion (Li-Ion) battery, and will be equipped with an X-band transponder and antenna for communication purposes.

ST-5, as part of the New Millennium Project, will aim to validate methods of operating multiple spacecraft as a system while testing new technologies near the Earth's magnetosphere. The technologies to be tested are as follows:

- Software that automatically operates each satellite and determines orbits.
- A communications system component that uses one-fourth the voltage and half the power of components used today. It also weighs 12 times less and is nine times smaller than proven technology.
- CMOS Ultra-Low Power Radiation-Tolerant (CULPRiT) technology that will be developed through transistor-level process modification that lowers internal turn-on threshold to near-zero levels to allow supply voltage operation as low as 0.5V. Dynamic biasing is then applied to adjust threshold for process variations. This technology uses 20 times less power than proven technology (Raphael, 2001).
- An electrically tunable coating that can change its properties from absorbing the Sun's heat when the spacecraft is cool to reflecting or emitting heat when needed.
- A very tiny micro electro-mechanical system chip that provides fine attitude adjustments on the spacecraft using 8.5 times less power and weighing less than half as much as proven systems.

- A rechargeable lithium ion battery that stores two to four times more energy and has a longer life than proven technology.

The technologies are all very new and, if successful, will make new missions more effective and less expensive (Savage, 1999).

2.5 Basics of Spacecraft Flight

Spacecraft are governed by the same rules as planetary motion, the same rules laid down by Isaac Newton and Johann Kepler hundreds of years ago. Kepler's First Law states that, "The orbit of each planet is an ellipse, with the sun at a focus." (Figure 1)

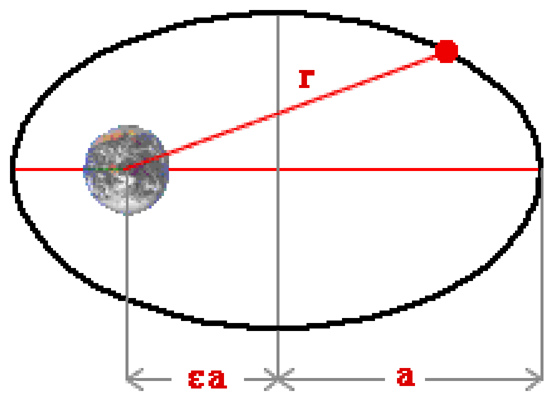


Figure 1: Kepler's First Law

(Guidry, 2002)

His Second Law states, "The line joining the planet to the sun sweeps out equal areas in equal times." (Figure 2)

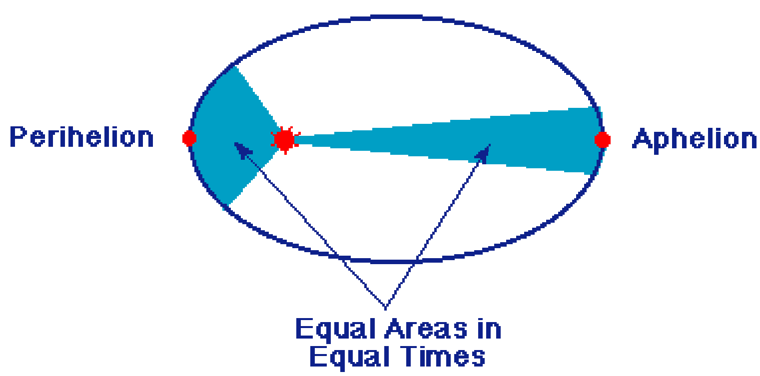


Figure 2: Kepler's Second Law

(Guidry, 2002)

Kepler's Third Law, "The square of the period of a planet is proportional to the cube of its mean distance from the sun (Bate, 1971)." (Figure 3) Each of these laws applies also to satellites and other spacecraft due to the laws of gravitation set forth by Newton; the behavior of a satellite orbiting a planet mimics that of a planet orbiting the sun.

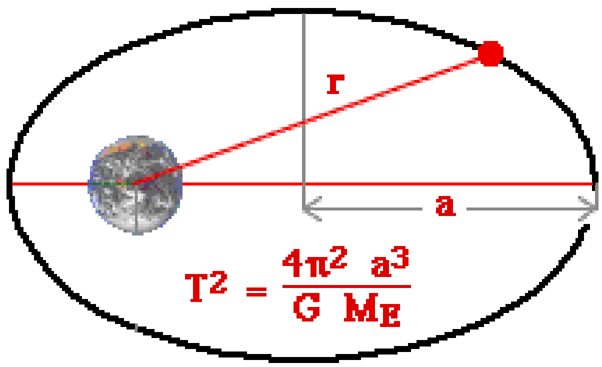


Figure 3: Kepler's Third Law

(Guidry, 2002)

The behavior described by Kepler's Laws is explained by Newton's Laws of Motion and his Law of Universal Gravitation. Newton's First Law explains that, "Every body continues in its state of rest or of uniform motion in a straight line unless it is compelled to change that state by forces impressed upon it." His Second Law, "The rate of change of momentum is proportional to the force impressed and is in the same direction as that force." His Third Law, "To every action there is always opposed an equal reaction (Bate, 1971)." A spacecraft orbiting a planet is essentially in perpetual free-fall around that planet; a launch vehicle is used to provide a spacecraft with enough energy to attain an altitude that allows it to circle the earth (or other planetary body) and it is then held in place by the centripetal force exerted by gravity, resulting in a perpetual, elliptical free-fall (Basics of Space Craft Flight, 2002).

The specific ellipse that the spacecraft is following is the orbit referred to in Kepler's Laws.

There are seven elements that are necessary to mathematically describe an orbit (Figure 4):

semi-major axis (a) The semi-major axis is one-half the maximum diameter, or the distance from the center of the ellipse to one of the far ends.

eccentricity (e) Eccentricity is the determination of the exact shape of the ellipse; for every ellipse, there are two fixed points, called foci, such that the sum of the distance from any point on the perimeter of the ellipse to the foci is always constant. The eccentricity of an ellipse is the distance between the foci, divided by the length of the semi-major axis.

inclination (i) Inclination is the angular distance from the plane of the earth's equator to the plane of the orbit.

argument of periapsis (ω) The argument of the periapsis is the angular distance from the periapsis (the point in orbit which is closest to the earth's surface) to the ascending node (point where the orbit passes through the ecliptic plane, going North) (*Basics of Space Craft Flight,* 2002).

longitude of the ascending node (Ω) The longitude of the ascending node is the angle between the line within the equatorial plane pointing toward the vernal equinox and the line formed by the intersection of the equatorial orbital planes, where the orbital motion is ascending (from south to north). The longitude of the ascending node describes the rotation of the orbital plane.

argument of perigee in degrees (w) the angle between the ascending node and the orbit's point of closest approach to the earth (perigee)

true anomaly in degrees (v) the angle between perigee and the vehicle (in the orbit plane) (Campbell, 1996).

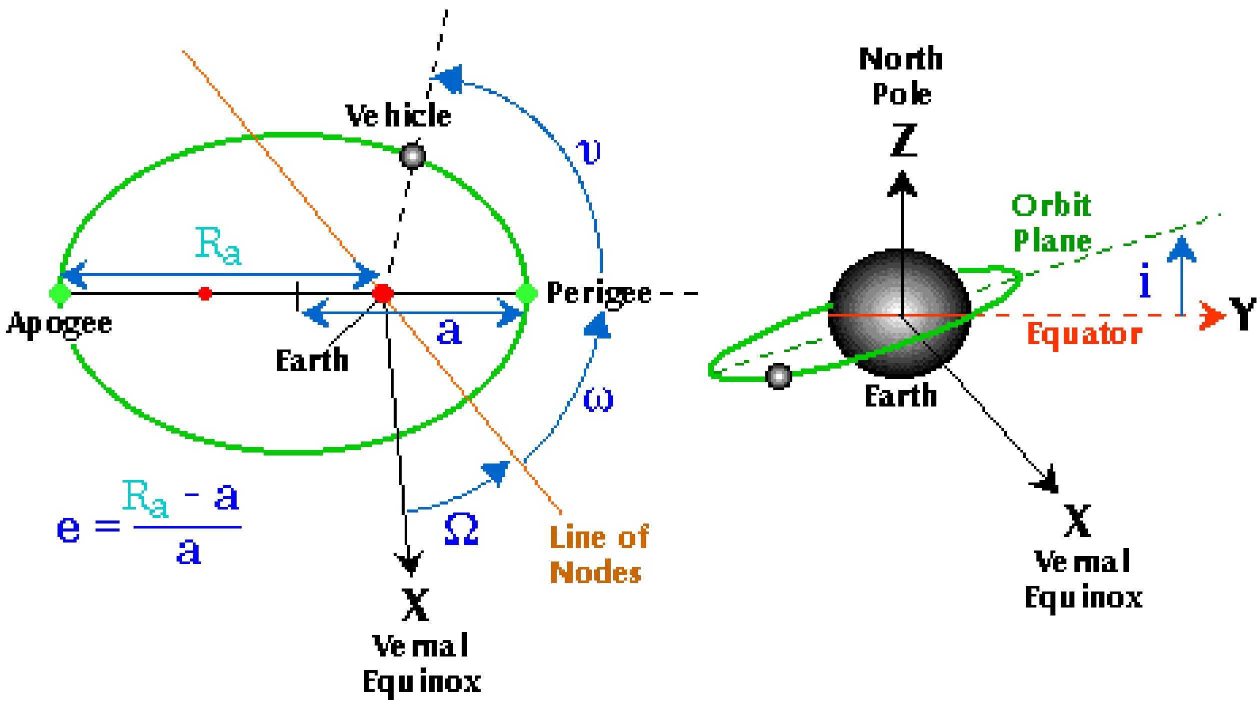


Figure 4: Orbital Elements

(Exploration, 1995)

2.6 Common Earth Orbits

There are several different earth orbits that spacecraft commonly use. Satellites in Low Earth Orbit (LEO) are 200 – 500 miles above the earth's surface and travel at approximately 17,000 miles per hour, allowing them to completely orbit the earth in about 90 minutes (The Tech, 2002). Satellites in Geosynchronous, or Geostationary, Equatorial Orbit (GEO) are located 22,300 miles above the equator and take a full 24 hours to orbit the earth; the equatorial location and orbit period cause the satellite to appear stationary with respect to the earth (The Tech, 2002). When a GEO satellite is launched, it is first launched into an elliptical orbit, called a Geosynchronous Transfer Orbit (GTO), and then uses a rocket engine to circularize the orbit at the equator (*Basics of Space Craft Flight, 2002)*.

2.7 Power Sources

On orbit, most modern satellites use solar panels to generate electricity. Solar panels are collections of solar cells that absorb energy from the sun and convert it to electricity. Some of the more efficient solar panels use solar concentrators to convert approximately 20% of their absorbed energy to electricity (Small Satellites Home Page, 2002). Satellites also have batteries, which are used to store the generated electricity that is not immediately used. The batteries are drained during eclipse periods, when the solar panels cannot generate electricity, and during peak power usage periods when the power being used exceeds the power provided by the solar panels. The most common satellite batteries are Nickel Cadmium and Nickel Hydrogen, but Li-Ion batteries have been flown on three (3) small satellites thus far and appear a promising alternative to the more traditional batteries (Small Satellites Home Page, 2002).

2.8 ST-5

2.8.1 ST-5 Orbit

The ST-5 spacecraft will be processed at Kennedy Space Center (KSC) for launch as secondary payloads on an expendable launch vehicle bound for Geo-stationary Transfer Orbit (GTO). The baseline Geo-synchronous transfer orbit will have the following parameters:

Perigee Altitude = 185km

Apogee Altitude = 35,786km

Inclination b/w 0° and 28.5°

The spacecraft will have autonomous Sun acquisition upon deployment. Each satellite's spin axis will cause the solar panels to be perpendicular (+/-5° to the Sun. It is also known that the first few orbits of flight will be completed in approximately 1 to 2 days. The attitude and orbit determination and maneuver planning consists of 2 to 5 orbits in approximately 3 days, in order to stabilize the satellite. The constellation geometry of ST-5 will be a string-of-pearls formation as graphically depicted at apogee in Figure 5 (Bibyk & Odendahl, 2001).

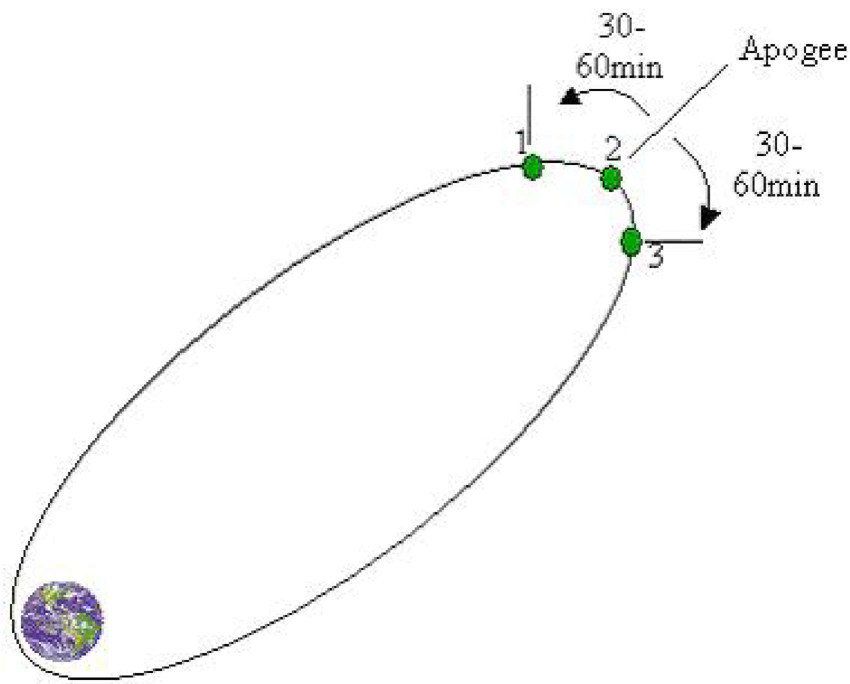


Figure 5: Baseline Geometric Configuration at Apogee

2.8.2 ST-5 Subsystems

Each ST-5 satellite can be divided into eight (8) subsystems that are necessary for the proper functioning of the satellite. The *structural/mechanical subsystem* is the physical satellite itself. The body of the satellite and all moving parts that are necessary for the functioning of the instruments and other subsystems are the components of the structural/mechanical subsystem. The *thermal subsystem* of ST-5 regulates the temperature of the satellite and of the various components within the satellite. ST-5's thermal subsystem includes the Variable Emittance Coatings (VEC), which are being flight-validated as part of this mission. The *electrical subsystem* handles the distribution of power to the other subsystems; the power is generated by the electrical power subsystem, consisting of solar arrays and a Li-Ion battery. The *radio frequency (RF) communications subsystem* allows the spacecraft to communicate with the ground and contains an X-band transponder, low noise amplifier (LNA), high power amplifier (HPA), and two (2) antennas (one mounted on the top of the spacecraft and a smaller one mounted on the bottom of the spacecraft). The *guidance, navigation, and control subsystem* uses a Miniature Spinning Sun Sensor (MSSS) and a magnetometer (mag) to gather data to determine the attitude of the satellite and ensure spin-stabilized control. The *propulsion subsystem* has cold gas micro-thrusters, which are fired in pulses to keep the satellite in orbit and at the appropriate distance from the Earth. Finally, the *Command and Data Handling (C&DH) subsystem* tracks and records data gathered by the telemetry devices and the magnetometer, monitors the status of the other subsystems, processes all flight software and operational commands, and controls communications with the ground stations (Stewart, 2002)

The three (3) subsystems that are pertinent to the behavioral model that was developed are: the electrical power subsystem, the data recorder (part of the C&DH subsystem), and the communications subsystem.

ST-5 Power Subsystem

The Electrical Power Subsystem (EPS) of ST-5, shown in Figure 6, consists of the Power System Electronics (PSE), mission unique solar array, and a Lithium Ion (Li-Ion) battery. For charge control, the PSE will employ solar array regulating electronics that will regulate the bus voltages generated by the solar array and the Li-ion battery.

The spacecraft is primarily powered by the solar array, which supplies the satellite with power through a boost regulator during phases of the orbit in which the spacecraft is illuminated by the sun. Power not used immediately by spacecraft subsystems is used to charge the battery. As the battery charges, the voltage it applies to the bus increases. If the battery is allowed to charge beyond a certain level, it can be damaged. Since the voltage on the bus is directly related to the battery state of charge, the bus voltage can be used to as a reference to regulate the amount of current that is flowing into the battery.

The regulator circuit included in ST-5 uses the bus voltage as a reference to regulate the amount of current that flows into the battery. If the voltage is too high, the circuit will limit the current flowing into the battery by creating a short circuit between the solar arrays and ground using a MOSFET. If the bus voltage is too low, the MOSFET will remain open allowing all available current to flow into the battery. This regulator circuit is governed by a pulse width modulator (PWM).

Pulse-Width Modulation: Modulation in which the duration of pulses is varied in accordance with some characteristic of the modulating signal (Bandwidth Market, 2002).

In the case of this specific PWM, the modulating signal is a 100 kHz triangle wave that modulates the error signal (difference between the bus voltage and the maximum voltage).

The energy stored in the Li-Ion battery will provide power to the spacecraft loads during the launch (post-separation), eclipse, and if needed, the peak load conditions during the sunlight mission phases. The battery used in its operational configuration will have a depth of discharge of 60% (the functional depth of discharge of the battery can go much lower, but will not be used below this threshold while in flight) with minimum charge duration of 8 hours per cycle and an operating temperature range of -10° to 40°C while in orbit (Stewart, 2002). While the satellite is in eclipse, all subsystems except the Command and Data Handling (C&DH), the transponder (in receive mode only), the Miniature Spinning Sun Sensor (MSSS), and possibly the magnetometer will be inactive.

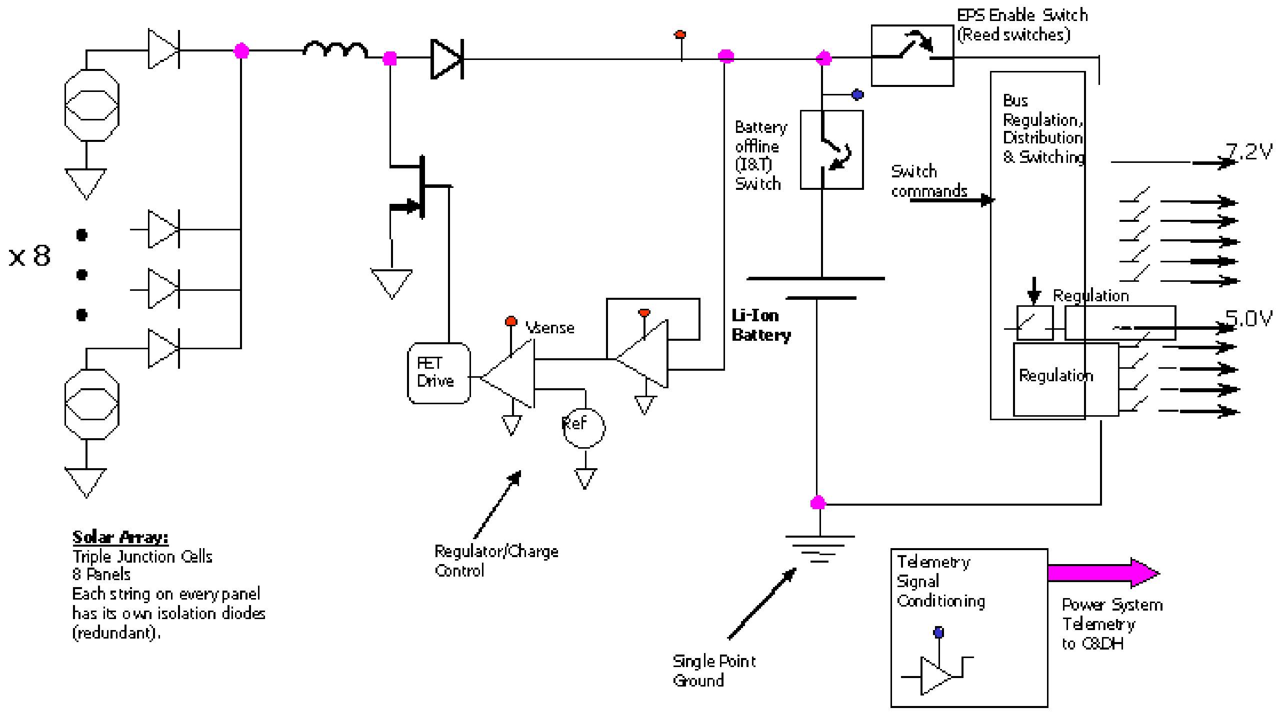


Figure 6: Functional Block Diagram of ST-5 EPS

The distribution of power from the EPS is pictured in Figure 7. Bus voltages distributed to spacecraft components include: an unregulated 7.2 V dc +/-1.2V bus, a regulated 5.0 V dc +/- 2% bus voltage to the Command and Data Handling (C&DH) system, and 5.23 V dc +/- 2% to the Thermal Control Electronics (TCE) and thermal technologies (Variable Emittance Coating). The power system will provide bus protection by monitoring bus voltage, battery voltage, battery state of charge, and total bus current. The power subsystem will also allow the C&DH and Flight Software (FSW) to send commands to the PSE in order to shed loads if overcurrent occurs. Ground command systems will have remote control over the removal and restoration of power to the loads.

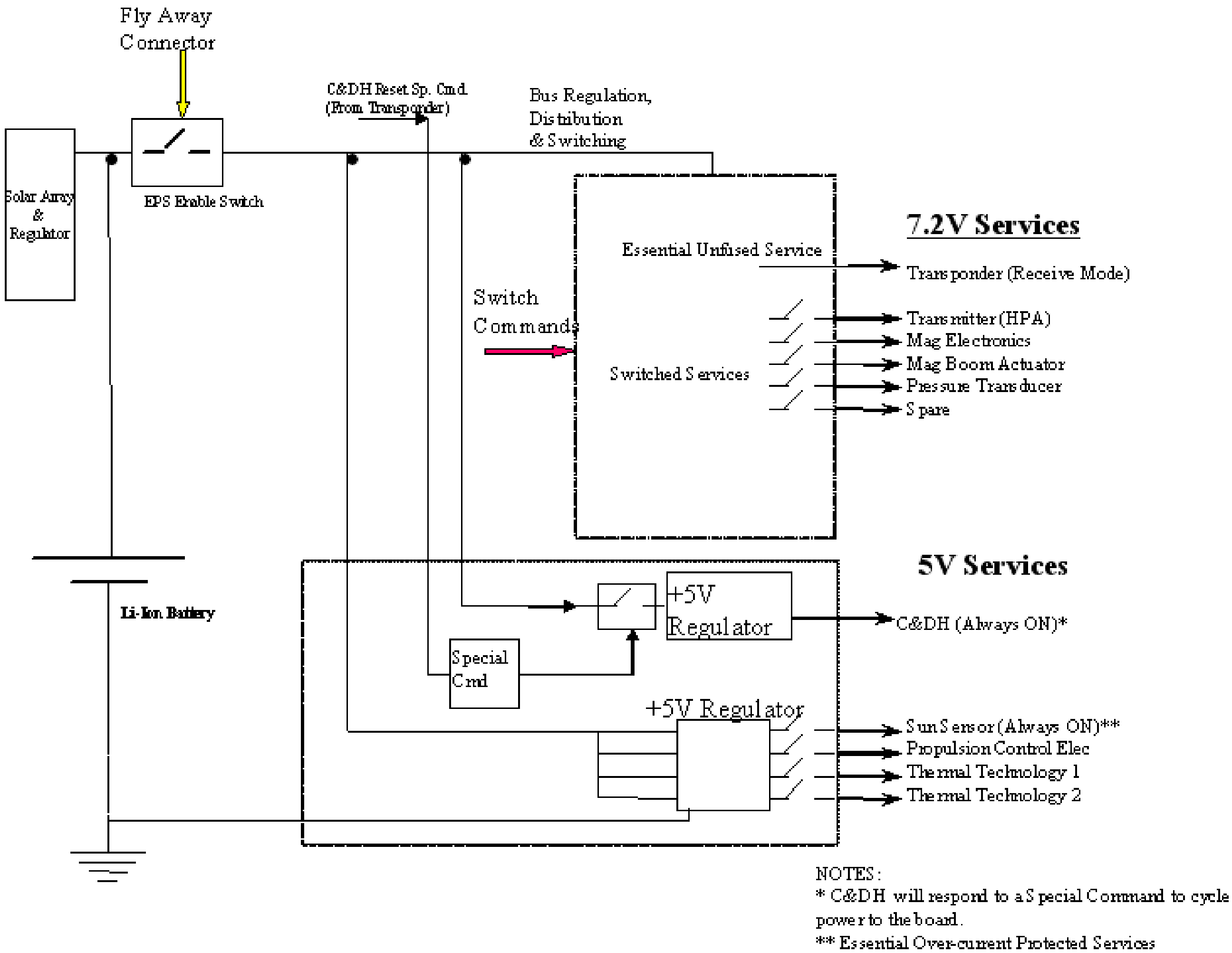


Figure 7: ST-5 Power Distribution

ST-5 Data Recorder

An integral part of any satellite is the data it collects. The data recorder must be able to collect data, store the data, and then transmit it back to Earth so it can be analyzed. In support of these functions, ST-5 incorporates a data recorder system and a transponder with a high power amplifier to transmit the data back to Earth.

As with all systems on the satellite, the data recorder (part of the C&DH subsystem) has a set of specifications that will determine how it is used. The data recorder has a planned capacity of 20 megabytes of solid-state memory that will be partitioned into three sections: the health and safety data of the satellite, the science data collected by the magnetometer, and real-time event data (ST-5 CDR, 2002).

The health and safety data will be stored in 9.96 megabytes of the data recorder's memory, which is assumed to be large enough to contain two complete orbits of telemetry data. The first orbit is expected to last 10.5 hours, and approximately 4.94 megabytes of data will be collected during this time. The 4.94 megabytes is almost 50% of the total capacity allotted to the health and safety data. If the second orbit's health and safety data is of a similar size, there is a possibility that the data recorder will not have the capacity to hold data from two (2) orbits, resulting in an overflow and loss of data (ST-5 CDR, 2002).

The data recorder also contains partitioned memory to store the science data collected by the magnetometer. Science data is collected at about 2.31 megabytes per orbit. Since ST-5 is not primarily a science mission, the science data is not considered as important as the health and safety data, thus science data overflow will result in an overwrite of previously collected science data (ST-5 CDR, 2002).

Real-time event data is also collected in the spacecraft's data recorders. The real-time data is only collected when a command is received from the ground station indicating to collect such data. Overall health and safety data and science data is transmitted to the ground separate from the real time data.

ST-5 Communications Subsystem

The ST-5 communications subsystem (Figure 8) contains an X-band micro-transponder, a diplexer, antennas, a low noise amplifier (LNA), and a high-power amplifier (HPA) for transmissions. The communications subsystem is intended for use with the Deep Space Network (DSN) ground stations (ST-5 CDR, 2002).

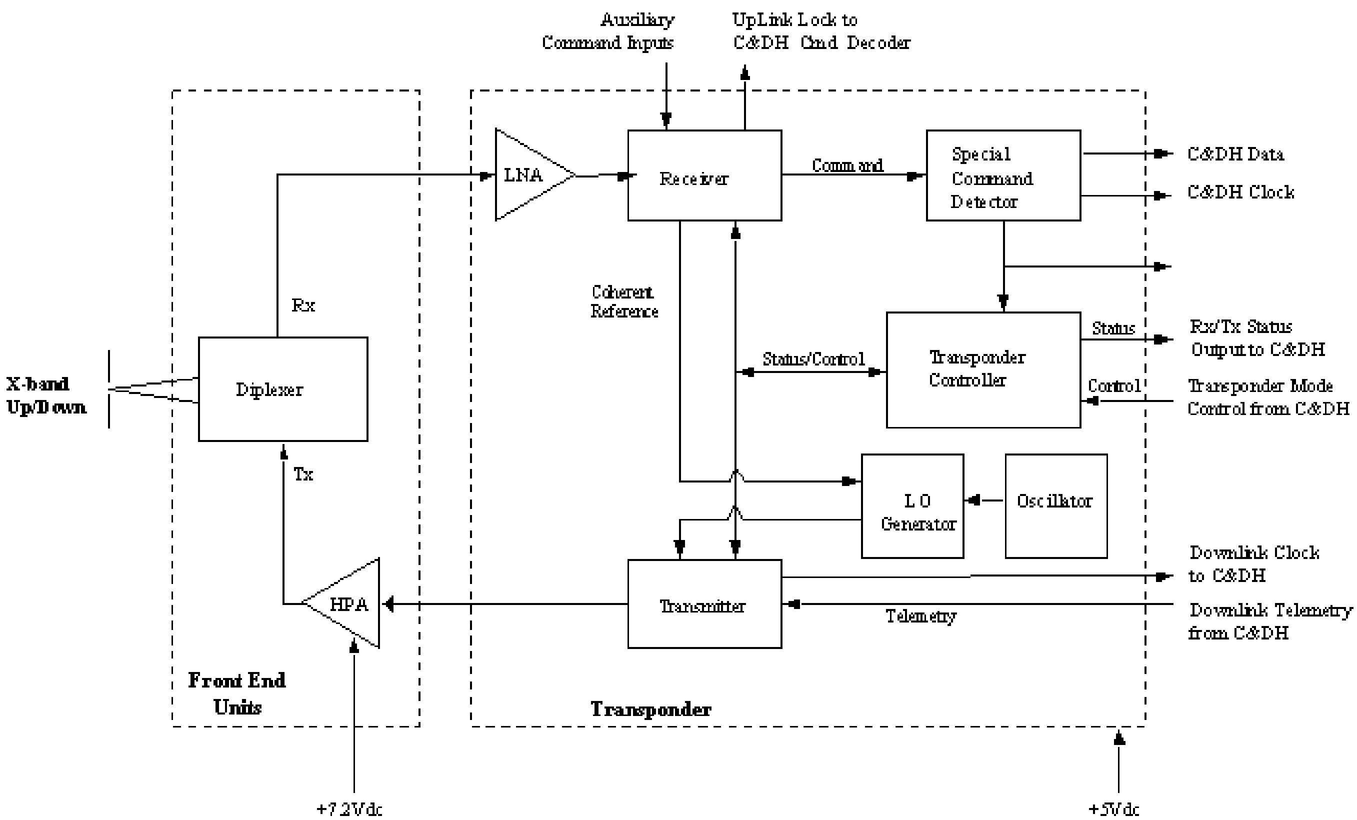


Figure 8: ST-5 Communications Subsystem

The X-band transponder is a NMP technology that is being flight-validated on this mission. The transponder has an uplink data rate of 1 Kbps and a downlink rate of 100 Kbps (an emergency, real-time downlink rate of 10 Kbps is also possible). Transmission frequency is ~ 8.5 GHz and transmissions will be accomplished at least once per orbit. The ST-5 satellites will be communicating with the Deep Space Network (DSN) ground stations; because DSN is a shared asset, the time and duration of transmission will be scheduled ahead of time, according to the projected orbit of the ST-5 satellites and the schedule of the DSN stations. Approximately 30 minutes per satellite per orbit will be allotted to transmit the data stored on the data recorder as well as the real-time housekeeping data that is transmitted. Due to the power constraints of the satellite, transmissions will not occur during eclipse periods, or when the magnetometer is collecting attitude data (ST-5 CDR, 2002).

2.9 Simulink Software

Simulink is part of the MATLAB software package distributed by The MathWorks, Inc. that can be used for modeling, simulating, and analyzing dynamic systems. The software allows for the modeling of linear and non-linear systems in continuous time, sampled time (single-rate or multi-rate), or both. Simulink provides a graphical user interface (GUI), which allows the user to build models as block diagrams, a revolutionary method of modeling that does not require the user to formulate differential and difference equations in a language or program (Dabney & Harman, 2001)

The models are hierarchical so that the user can use either a top-down or bottom-up approach, allowing for comprehensive organization of systems, subsystems, and components at different levels. This also provides the user with insight into how a model is organized and how the different parts interact. Once a model is designed it is not difficult to simulate it by using the simulation menus that Simulink provides, or by using the commands in MATLAB's command window. The simulation menu is useful for interactive simulations, while the command line is applicable for batches of simulations.

Simulink has a number of block libraries, which contain elements such as sinks, sources, linear and nonlinear components, and connectors. Simulink also allows the user to customize or create blocks. Customized blocks are created through the use of S-Functions, which serve to extend the capabilities of the Simulink software.

An S-Function is a way to use program logic in a Simulink block within a Simulink model. S-Function blocks consist of a set of inputs, a set of states, and a set of outputs. S-Functions are created based on an S-Function template, consisting of a top-level function and a set of skeleton sub-functions (S-function callbacks), provided with the Simulink software. The template is very general and easily modified to accommodate continuous, discrete, and hybrid systems. S-Functions can be created in MATLAB, C, C++, Fortran, or Ada; for this project, we used only MATLAB to create the functions.

Simulink was chosen for this project because it provides a simple and dynamic environment for the behavioral modeling. It also provides the designer with a comprehensive interface to allow for easy updating and tracking of inputs and outputs. The software is user-friendly, allowing the developed model to be run easily by users who do not necessarily know how the structure of the model works.

2.10 Summary

After years of satellites increasing in size, the focus is turning toward smaller, cheaper satellites. NASA's ST-5 project an example of this new focus. ST-5 is composed of three micro-satellites to be launched into a GTO orbit and flown for three (3) months in a simple formation. The mission of ST-5 is to verify new technologies in space flight. The critical systems for verifying the new technology are: the power subsystem, the data recorder, and the communications subsystem. In order to ensure successful completion of the mission, a behavioral model was created focusing on the critical subsystems using Simulink simulation software. The details of the behavioral model are described in the following chapter.

3. Modeling Methods

3.1 Introduction

The first step toward the creation of the ST-5 model was to integrate an existing orbital propagation model into the power model. The reason for doing this was to allow the user more flexibility when using the model; the user has the choice of inputting eclipse/illumination cycles from the orbit propagator model directly into the power model or of entering eclipse/illumination information manually into the operational scenario. The specific orbital location, also generated by the orbit propagator, is used as an input into the other subsystems, to determine the state of the subsystems.

Models of the Data Recorder and portions of the Communications System were then created and integrated with the updated power model. The existing power model was also modified to allow for the simulation of a solar array failure. The voltage regulator portion of the power model was then modeled, separately, to test the effects of pulse-width modulation of the signal. Finally, pre-existing, planned operational scenarios were incorporated into the model.

3.2 Model Components

3.2.1 Voltage Regulator

The voltage regulator controls the power to the spacecraft loads, controls the charging and discharging of the battery, and shunts any excess power generated by the solar array. Previously, the voltage regulator was modeled as a PID (Proportional, Integral, and Derivative) Controller that used an average feedback loop to regulate the bus and battery voltage, along with the actual solar array current (Figure 9). A second voltage regulator model was created to test the effects of a more realistic representation of the regulator by incorporating a pulse-width modulator and a filter (Figure 10).

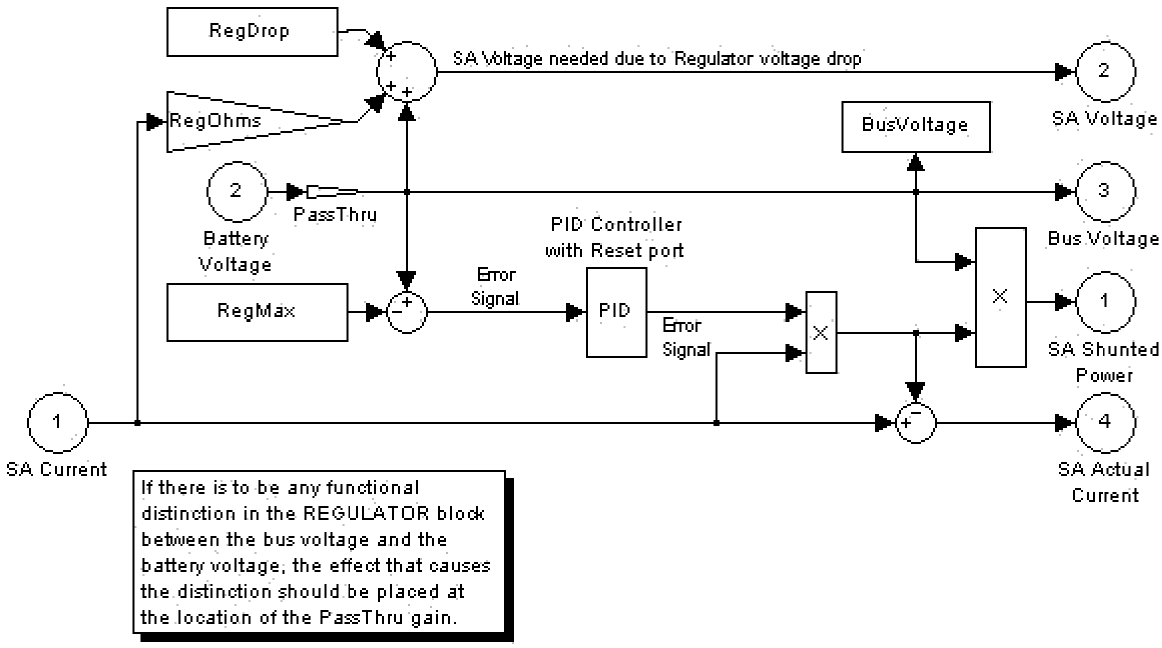


Figure 9: Original Voltage Regulator Model

The voltage regulator is a Direct Energy Transfer (DET) system, which means that the solar array acts as a current source when it is in the sun, feeding all of the spacecraft loads and charging the battery when excess power is available. The voltage regulator monitors the energy transfer and shunts excess energy, ensuring that the battery does not over-charge. The mechanism that switches the regulator to shunt or not shunt solar array power is a low-resistance MOSFET, which is driven by an integrator that compares the battery voltage to a reference voltage (8.4V) and generates an error signal. The generated error signal is fed into a comparator that acts as a pulse-width modulator, feeding directly back into the MOSFET gate.

The pre-existing regulator model did not contain a representation of the pulse-width modulation of the signal (Figure 9). The task was to create a voltage regulator model that incorporated the pulse-width modulation (PWM). The PWM compares the differential between the bus voltage and the maximum battery voltage to a 100 kHz triangle wave. The output signal of the PWM is the error differential modulated over a triangle wave, which serves to pass a clean square-wave into the MOSFET.

A more realistic representation of the voltage regulator was modeled in Simulink (Figure 10). The model incorporated the pulse-width modulation used in the EPS of the spacecraft and utilized the PID controller as an integrator, as the original model did, but compared the PID output signal to a triangle wave input from the Signal Generator. This comparison is completed in the shaded block and serves to model the pulse-width modulation.

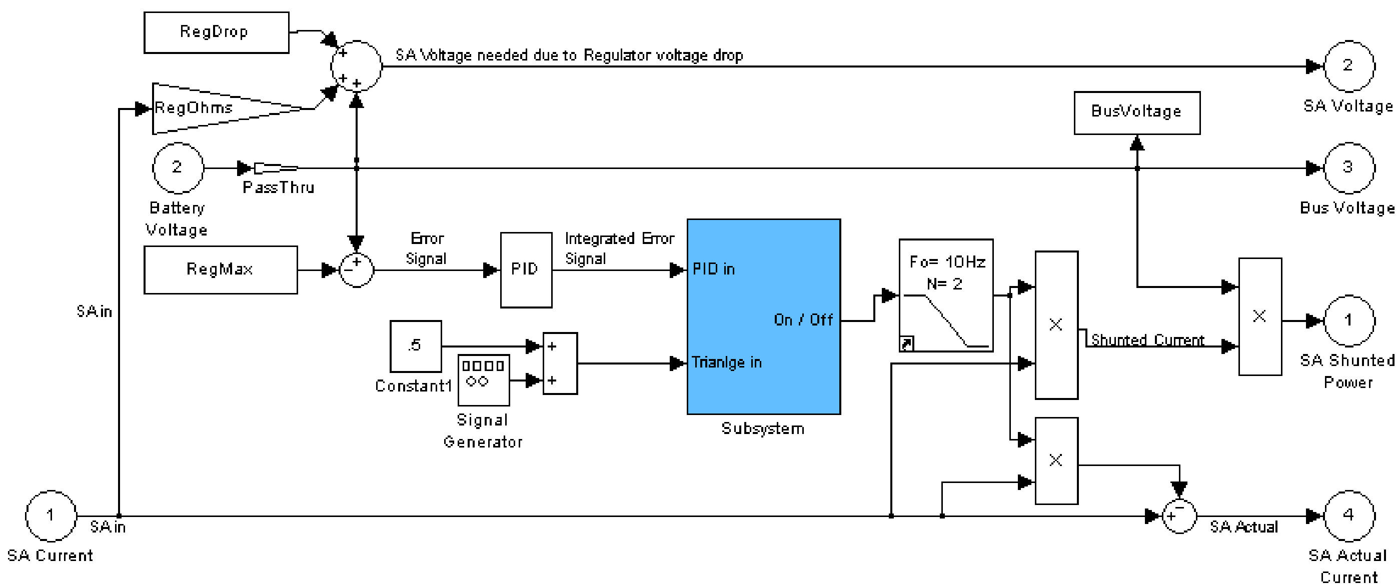


Figure 10: PWM Voltage Regulator

The pulse-width modulated signal output is fed into a filter, which serves to smooth the signal, eliminating very large jumps in the bus voltage and across the battery, which could harm the power system. The filter was modeled as a Butterworth Filter with a center frequency at $1/10^{th}$ (10 Hz) of the frequency of the PWM (100 Hz, for simulation purposes, based on the limitations of the simulation software).

3.2.2 Data Recorder Model

The data recorder model was primarily developed as an S-Function that is used to track the amount of data in the recorder at any given time (Figure 11). The inputs for this model are the telemetry rate, magnetometer rate, transmission times, the transmission rate, and the maximum storage capacity (pre-determined). The previously listed items are setup as variables in the workspace when an operational scenario is run.

The S-Function is the backbone of the data recorder model (located in the middle of Figure 11 labeled SFunData2). The S-Function serves to keep track of the amount of data stored in each section of memory. The model simulates the storing of data into five (5) different variables: Current Data, Previous Data, Total Data, Data To Be Transmitted, and Science Data. The listed variables are tracked in the S-Function under several "if" statements that set the conditions to determine the transmission of data and transfer of data from one variable to another. For example when a transmission is initialized the amount of data in Previous Data is dumped, therefore eliminating that data set from Total Data, the Current Data is transferred to Previous Data, and the Data To Be Transmitted is set equal to the sum of the Current Data, Previous Data, and Science Data. During transmission the Data To Be Transmitted is also being adjusted according to time steps, by subtracting out the rate at which the data is being transmitted (labeled Transmit Rate in Figure 11). At the transmission the Current Data and Science Data are also reset to zero (0), in order to begin collecting the next orbit's data. The S-Function also ensures that the values in the data sections never exceed the maximum amount of data by setting the limit to the Max Size, which is set in the operational scenarios; it also ensures that the Data To Be Transmitted never goes below zero (0).

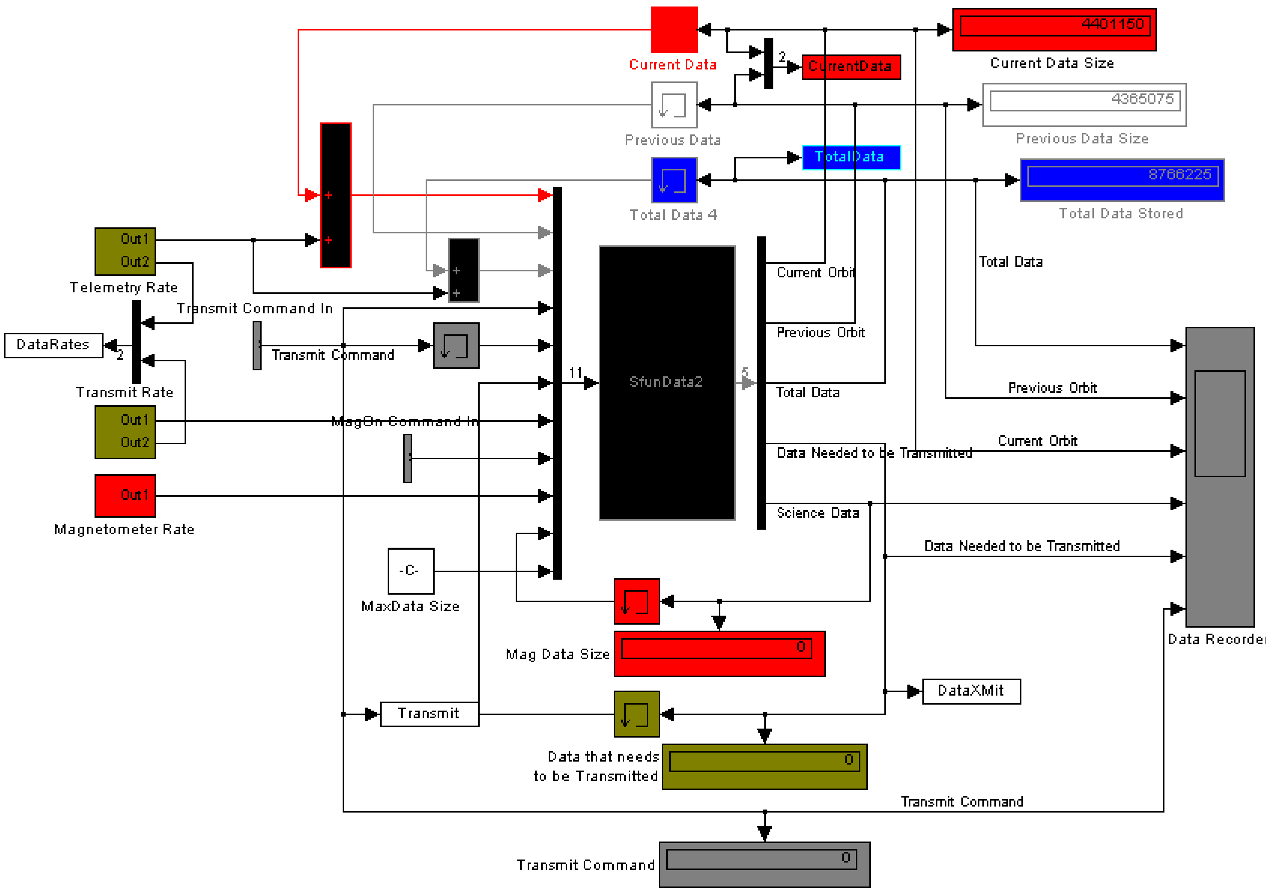


Figure 11: Data Recorder Simulink Model

The blocks labeled Telemetry Rate, Transmit Rate, and Magnetometer Rate are masks that calculate the amount each S-function variable is modified, based on each time step (Figure 11). All three masks are identical, except for the rate that is input from the operational scenarios (TelemetryArray, CommRate, MagnetometerArray). The Comm Rate is seen below in Figure 12, which shows how the three rates blocks work. Each block consists of the rate from the operational scenarios, a clock, and a memory block. The memory block holds the time of the previous time step and it subtracts from the current clock to multiply the Δt by the rate to calculate the data collected at each timestep.

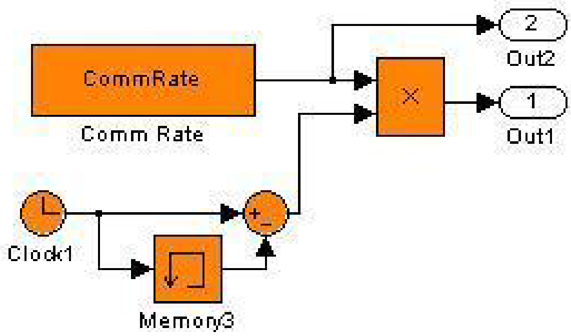


Figure 12: Comm Rate

The behavioral model of the health and safety and science partitions of the data recorder system must take several things into account. It must be able to interpret the rate of data collection, the time between transmissions, and the downlink rate of the transponder. Our model, built in Simulink, takes all these factors into account and can be used to determine whether or not data overflow occurs and whether there is a sufficient downlink window to transmit all the data to the ground station. This model can be used to test operational scenarios and alert the user of any possible overflows or errors in the data recorder memory.

3.2.3 Communications Model

The communications model developed for the ST-5 satellites (Figure 13) is used to determine the link margin, the Doppler shift seen from the ground stations, and which ground stations the satellite can communicate with at the times of desired transmissions. An S-function, *SFunGStations*, is used in this model to determine the ground station look angles necessary to view the satellite. Subsystem blocks, located within the *Demux and Remux* subsystem, are used to calculate the link margin and Doppler shift for each ground station. Finally, the *Line of Sight Logic* is used to determine which ground stations the satellite can communicate with at a given time.

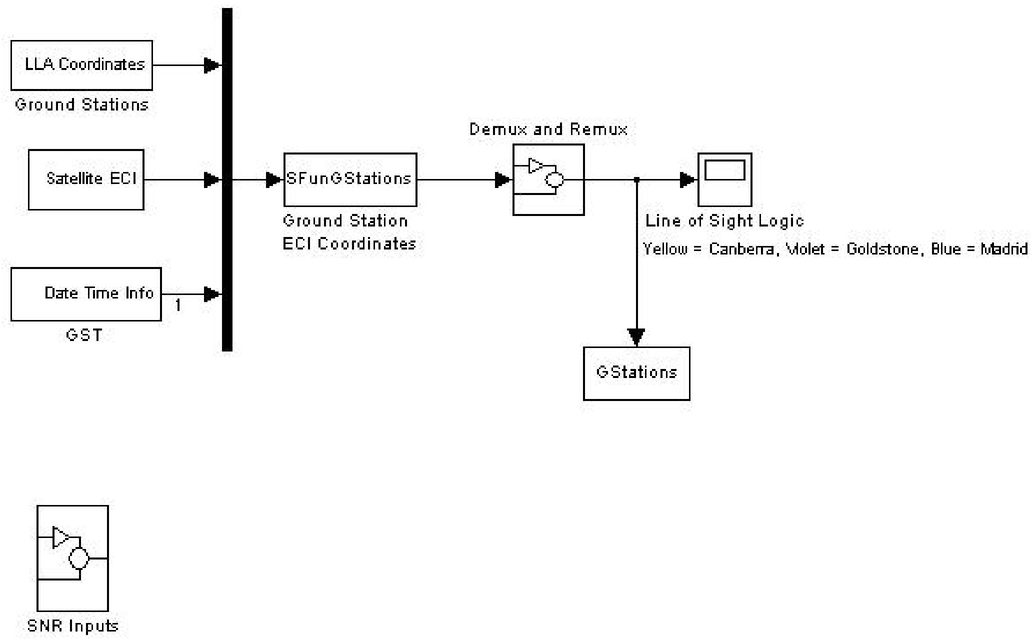


Figure 13: Top-Level Communications Block Model

<u>Inputs and Outputs</u>

Prior to running the communications model, an initialization file (i.e. Operational Scenario) and the orbit propagator model must be run. The initialization file contains constant inputs: the latitude and longitude of each ground station, the date and time that the simulations starts, and the antenna constants used for the link margin calculation. The latitude and longitude information is imported into the *Ground Stations* block, the date and time information is imported into the *GST* block, and the antenna inputs are imported into the *SNR Inputs* block (Figure 13).

The orbit propagator model generates the satellite ECI position as a function of time and outputs the position information as an array. The satellite ECI position array is then imported into the *Satellite ECI* block (Figure 13), used within the S-Function, *SFunGStations*, to determine the range and angles between the satellite and the ground stations.

With these inputs, the communications model outputs an array showing each ground station and a logic 1 (can transmit) or 0 (cannot transmit) as a function of time. The communications model also generates a graph with this information, where a logical high indicates ability to transmit and a logical low indicates an inability to transmit (Figure 14).

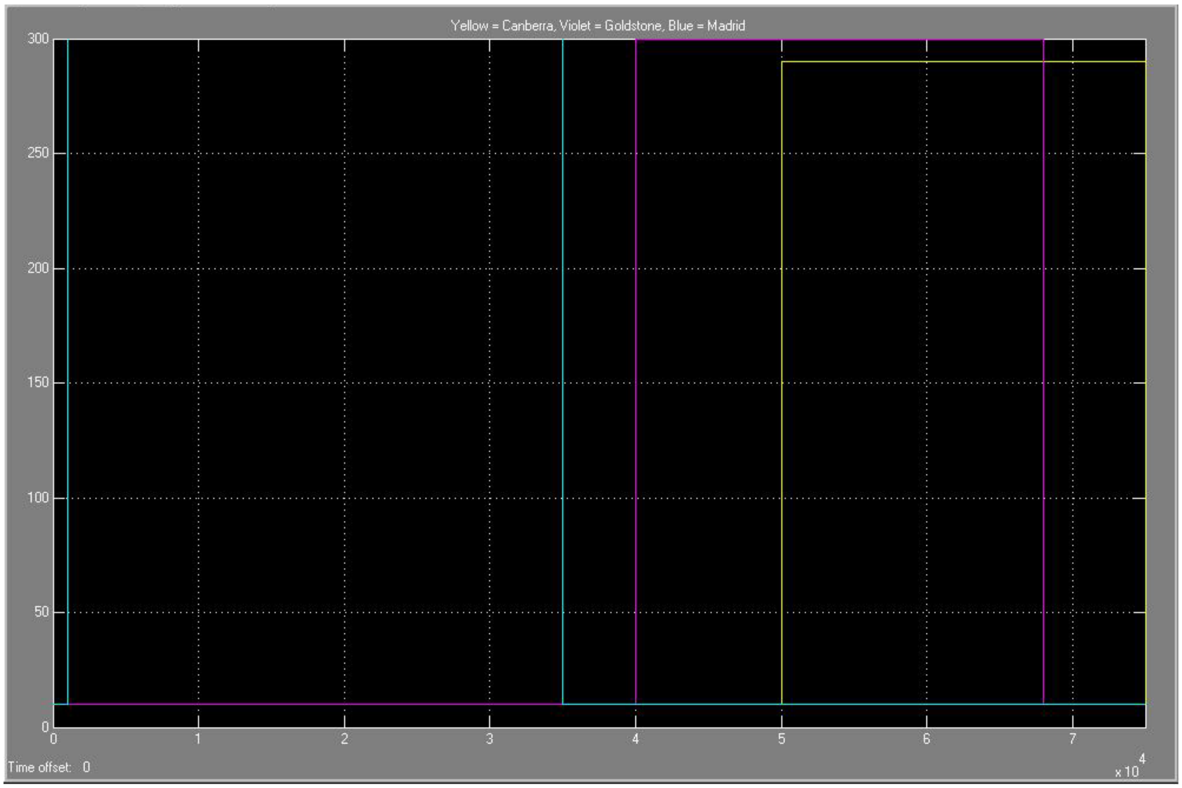


Figure 14: Ground Station Ability to Transmit - Communications Block Output

<u>Link Margin Analysis</u>

Link margin is the difference between the required signal to noise ratio and the actual signal to noise ratio. Link margins that are too low can result in unacceptable bit error rates in the received signal. The required link margin varies greatly depending on the specifics of the mission and can be considered, in part, a function of frequency. Generally, higher frequencies (above 10GHz) require more link margin, to account for greater atmospheric attenuation.

A link margin calculation begins with a calculated signal to noise ratio. The calculated signal to noise ratio takes into account the antenna power and gain, the data rate, the frequency, and the various losses that would affect the signal, such as free space loss, atmospheric losses, and passive losses (loss occurring in the circuit elements between the transmitter and the antenna). The equation for the signal to noise ratio is:

$$\frac{E_b}{N_0} = EIRP + L_s + L_a + G_r - T_s + 228.6 - R$$

where EIRP is the Effective Isotropic Radiated Power, L_s is the space loss, L_a is the transmission path loss, G_r is the receiving antenna gain, T_s is the system noise temperature (in Kelvin), 228.6 is Boltzmann's Constant, and R is the data rate. The transmitting antenna power and gain is accounted for in the EIRP, as is any loss occurring between the transmitster and antenna. All quantities are measured in decibels (dB) (Sklar, 1998).

The link margin calculator (Figure 15) for the ST-5 satellites takes as constant inputs all of the above quantities except the EIRP and L_s. L_s is calculated within a sub-block of the signal to noise calculation block, because it is a function of distance. EIRP is not calculated in this model; however, all elements of the EIRP are accounted for individually within the signal to noise calculation block. Once the signal to noise ratio is calculated, the Simulink model compares that value to the signal to noise ratio required for this mission (4.45 dB) and outputs the link margin.

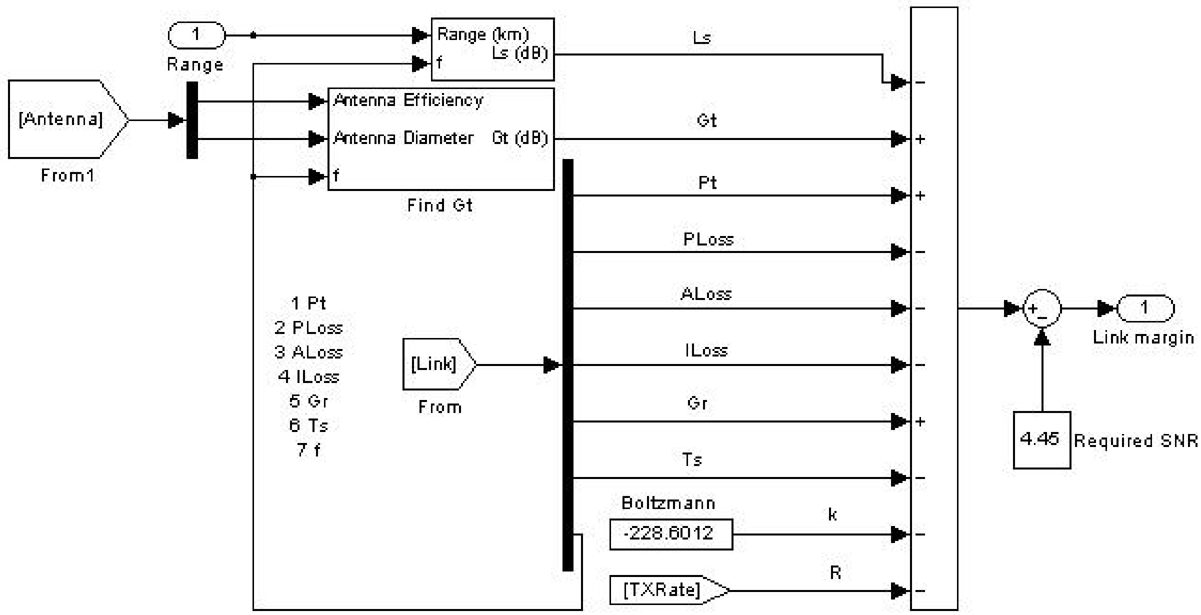


Figure 15: Link Margin Calculation Block for ST-5

For the ST-5 mission, the link margin is not expected to be a problem as the ground antennas that will be used are 34 meter-diameter Deep Space Network (DSN) antennas, with a gain of 68.34 dB. The choice of ground antenna ensures that the satellites will be able to transmit at any point in the orbit, except perigee. The satellites will not be able to transmit while in perigee because they are so close to the earth (less than 1 Re away) and are moving too quickly for the ground antennas to be able to track them.

Ground Station Visibility

The ability of the ground station to see the satellite depends on the location of the satellite in orbit and the maneuverability of the ground station antenna. Coordinates called "Look Angles" give angular measurements of where the ground station antenna needs to be pointed at a given time in order to see the satellite and communicate with it.

For the ST-5 model, the look angles are calculated within the S-Function, SfunGstations (Figure 13). The calculation can be broken up into four (4) parts: the Julian Date calculation, the Greenwich Sidereal Time calculation, the Latitude/Longitude/Altitude to

Earth Centered Inertial Coordinates conversion, and finally the Earth Centered Inertial Coordinates to Look Angles conversion.

Julian Date Calculation

Julian Day (JD) is a numerical representation of the date information represented by the Day/Month/Year format of the Gregorian Calendar. The time of day for a particular JD can be represented as a decimal fraction appended to the JD number. The decimal fraction .0 following a JD number indicates that the time of day is noon, as JD measures the day from noon to noon not from midnight to midnight (Sklar, 1998).

The JD numbers begin with a reference date at noon, November 24, -4713, Gregorian. The JD for noon, November 24, -4713 is 0. More recent JDs are used as references for counting JDs, and it is common to reference back to the first day of January of a given decade to simplify JD calculations. The equation used for the Julian Date calculation:

$$J_{date} = fix(365.25 * (year + 4716)) + fix(30.6 * (month + 1)) + day + 2 - fix(year/100) + fix(year/400) - 1524$$

is taken from Astronomical Algorithms by Jean Meeus.

Greenwich Sidereal Time

The Greenwich Sidereal Time (GST) is the angle between the Prime Meridian and the Vernal Equinox. The GST is used to calculate the local sidereal time for a given latitude, which is necessary to calculate the Earth Centered Inertial (ECI) coordinates for a given point on the earth's surface.

The GST calculator used in the model takes a Julian Date and time of day (hour, minute, second) input and converts it into a single number. The conversion into a single number is done by first subtracting 0.5 from the JD (so that the interval measured is from midnight to midnight, not noon to noon), and then adding as fractions the exact time of day. The equation used to obtain the single number representation of the JD and time is:

$$Current = JD - 0.5 + hr * 0.0417 + \min ute * 6.94e^{-4} + \sec * 1.156e^{-5}$$

After converting the current date and time into a single number, subtraction is used to determine the amount of time passed since a certain reference date (JD 2440952.5 or midnight, January 1, 1971). The GST, in radians, is known for this reference date and can be calculated for any known date and time using the equation:

$$\Theta_g = \Theta_g 0 + 1.0027379093 * 2 * \Theta * D$$

Where Θ_g is the GST in radians; $\Theta_g 0$ is the GST in radians of the reference date; 1.0027379093 is the number of rotations the earth completes in one solar day; and D is the difference between the reference date and the current date (Bate, 1971).

<u>Latitude/Longitude/Altitude to ECI Coordinate Conversion</u>

The conversion from Latitude/Longitude/Altitude coordinates to Earth Centered Inertial (ECI) coordinates (Figure 16 and Figure 17) is a necessary intermediate calculation to determine the visibility of each ground station. Because the ECI coordinate system is fixed while the Earth rotates around it, the ECI coordinates of each ground station will change with time.

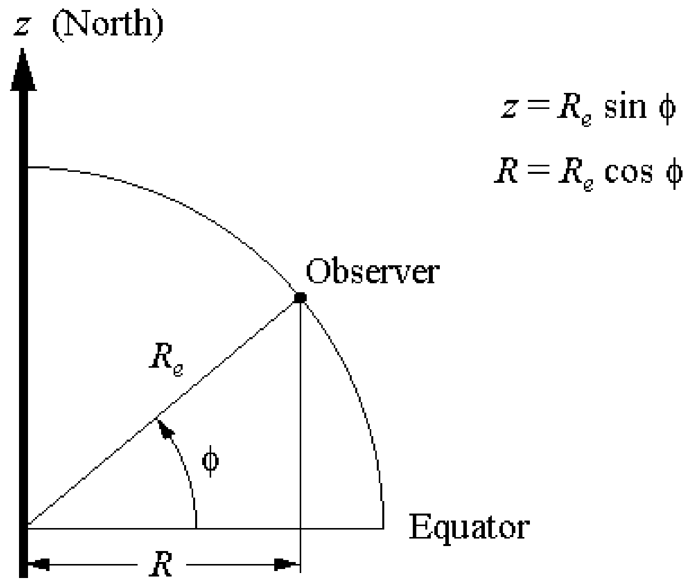


Figure 16: Latitude to ECI conversion

http://celestrak.com/columns/v02n01

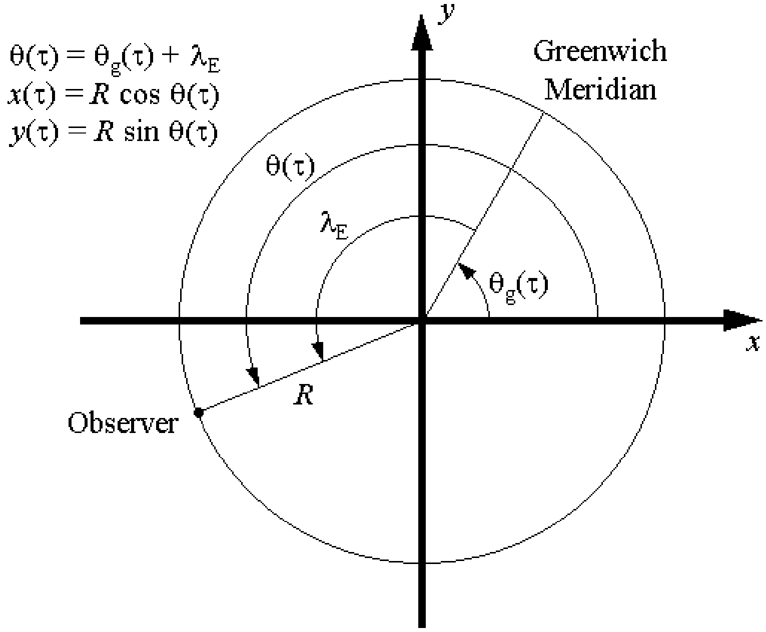


Figure 17: Longitude to ECI conversion

http://celestrak.com/columns/v02n01

The calculation method used in this model was used in the FalconSat behavioral model. The first step is the computation of the local sidereal time (LST), the angle between the Vernal Equinox and the local longitude. The local sidereal time is calculated by adding the Greenwich Sidereal Time to the local longitude. The second step is the computation of two (2) geodetic constants, which account for the flattening of the Earth. The final step is the use of the geodetic constants and the latitude and longitude to calculate the ECI coordinates at a given time.

The ECI coordinates are calculated as:

$$
\begin{aligned}
X &= c * \cos(latitude) * \cos(LST) \\
Y &= c * \cos(latitude) * \sin(LST) \\
Z &= d * \sin(latitude)
\end{aligned}
$$

where *c* and *d* are the geodetic constants, calculated as:

$$
\begin{aligned}
b &= \sqrt{\left(1 - \left(flat(2 - flat) * \sin(latitude)^2\right)\right)} \\
c &= R_{eq} / b + 0.001 * altitude \\
d &= R_{eq} * (1 - flat)^2 / b + 0.001 * altitude
\end{aligned}
$$

where *flat* is the flattening of the Earth and R_{eq} is the radius of the Earth at the equator.

<u>ECI to Look Angle Conversion</u>

Look angles are the azimuth and elevation necessary for a ground station antenna to see a satellite. Azimuth is the horizontal direction of the antenna, expressed as the angular distance between the antenna and the satellite. The azimuth is measured clockwise from North to South, so that 0° is North, 90° East, 180° South, and 270° West. Elevation is the angular distance of the satellite above the horizon in relation to the antenna, i.e. the up and down position of the antenna necessary to see the satellite. Look angle calculations require several steps, the first of which is the calculation of a range vector.

Once the ground station's ECI coordinates are known, the range vector can be calculated. The range vector is the difference between the satellite's ECI coordinates and the ground station's ECI coordinates.

The range vector is calculated as:

$$[r_x, r_y, r_z] = [x_s - x_g, y_s - y_g, z_s - z_g]$$

Where $[r_x, r_y, r_z]$ is the range vector, $[x_s, y_s, z_s]$ are the satellite's ECI coordinates, and $[x_g, y_g, z_g]$ are the ground station's ECI coordinates.

The range vector calculated is in the ECI coordinate system and must be converted to the Topocentric-Horizon (SEZ) coordinate system (Figure 18) in order to determine satellite visibility. The transformation to SEZ is done by rotating through the local sidereal time about the Z-axis (the Earth's rotation axis) and then through the ground station's latitude about the Y-axis. The equations used for these calculations are:

$$r_S = \sin(\varphi)\cos(\theta) * r_x + \sin(\varphi)\sin(\theta) * r_y - \sin(\varphi) * r_z$$
$$r_E = -\sin(\theta) * r_x + \cos(\theta) * r_y$$
$$r_Z = \cos(\varphi)\cos(\theta) * r_x + \cos(\varphi)\sin(\theta) * r_y + \sin(\varphi) * r_z$$

where r_S, r_E, and r_Z are the range vector expressed in SEZ coordinates, θ is the latitude and φ is the local sidereal time.

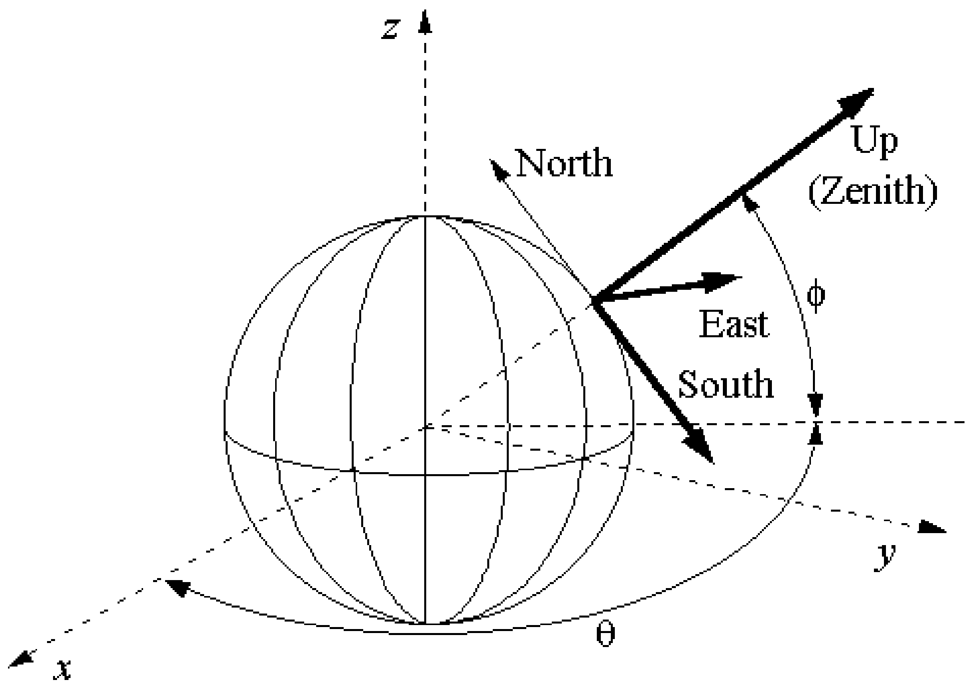


Figure 18: Topocentric-Horizon Coordinate System

http://celestrak.com/columns/v02n02

From the SEZ coordinates, the look angles can be calculated by:

$$range = \sqrt{(r_S^2 + r_E^2 + r_Z^2)}$$
$$Elevation = \sin^{-1}(r_Z / range)$$
$$Azimuth = \tan^{-1}(-r_E / r_S)$$

The limitations with regard to satellite visibility are dependant on the specific antenna used. For the ST-5 mission, the antennas being used have a minimum elevation requirement of 10° and no azimuth restrictions. So any time the elevation is between 10° and 170°, the satellite is visible.

Doppler Shift Calculation

Doppler is the apparent change in the frequency of a transmitted signal, as seen by the receiver. The transmitting frequency of the ST-5 satellites is 8.74 GHz; the frequency received by the ground station antennas will be 8.74 GHz plus/minus the Doppler frequency shift. The Doppler shift is calculated using:

$$\Delta f = \frac{v_{rel} * f}{c}$$

where Δf is the change in frequency, v_{rel} is the velocity of the spacecraft relative to the velocity of the ground station, f is the transmitted frequency, and c is the speed of light.

The ST-5 Doppler calculation block (Figure 19) takes a range input from the ground station S-Function, SFunGStations. As the satellite moves in orbit, the range changes. The change in range is calculated by subtracting the previous range from the current range. The range change is then divided by the amount of time the satellite required to move from the previous range to the current range, to determine the relative velocity (Figure 20). The relative velocity is then input into the Doppler Calculation block (Figure 21), which is a block representation of the equation given above.

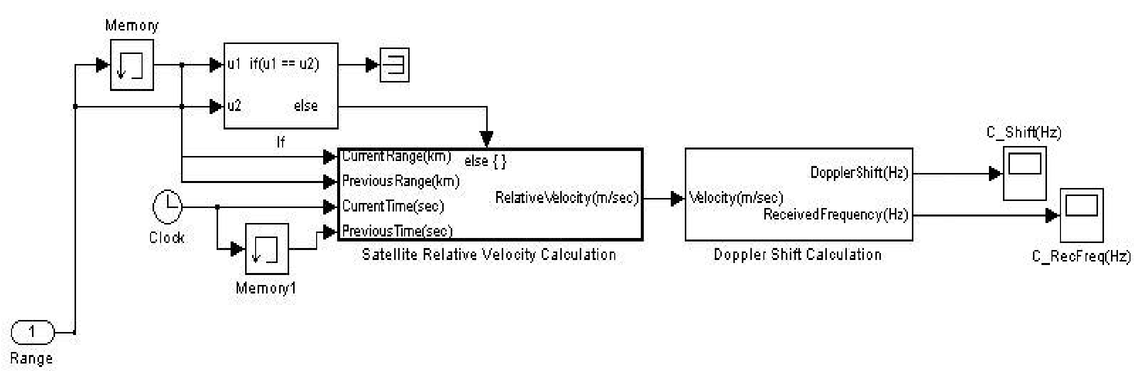


Figure 19: High-Level Doppler Calculation Subsystem

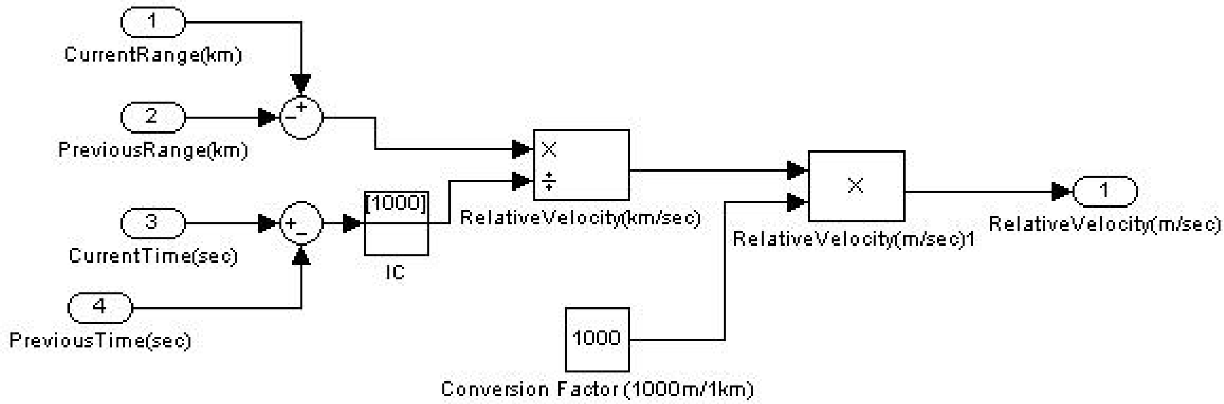


Figure 20: Satellite Relative Velocity Calculation Subsystem

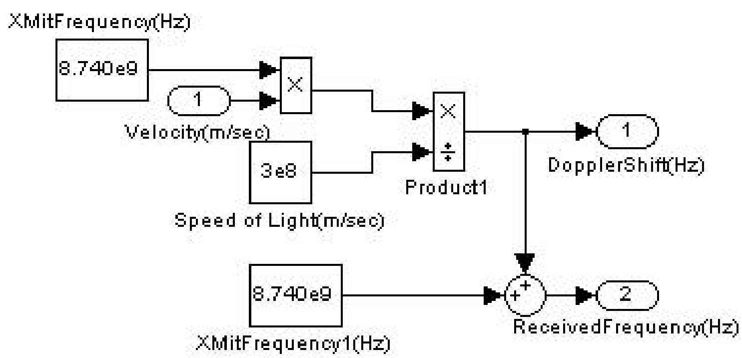


Figure 21: Doppler Shift Calculation Subsystem

<u>Communications Model Verification</u>

Each section of the communications model was built and tested individually, prior to being integrated into the communications model. This was done to verify that each section functioned correctly in a standalone capacity to reduce errors and confusion when the model was integrated.

The ground station look angle calculations were verified using a scenario given in "Orbital Coordinate Systems, Part III", by Dr T.S. Kelso, where the date was given as 18 Nov 1995, the time 12:46PM, ground site latitude was 45° N and longitude was 93° W, and the satellite ECI position was [-4400.594, 1932.870, 4760.712]. With these given values, Ds Kelso computed the Azimuth as 100.36° and the Elevation as 81.62°. The communications block, using the same given input values, calculates the Azimuth as 102.92° and the Elevation as 83.11°. The error in the Azimuth calculation is 2.55% and in the Elevation calculation is 1.83%. This error includes any error generated by the latitude/longitude to ECI coordinate conversion.

The ground station latitude/longitude to ECI coordinate conversion was verified using another scenario given by Dr Kelso, where the date was 01 Oct 1995, the time was 9AM, and the latitude and longitude were 40° N and 75° W, respectively. The ECI coordinates given by Dr Kelso were [1703.295, 4586.650, 4077.984]. The ECI coordinates calculated by the communications model were [1694.65, 4589.85, 4077.99]. The error in the ECI coordinates was [.51%, 0.07%, 0.00%] and the total position error was 9.22 km.

Link margin calculations were compared to an Excel worksheet provided by Victor Sank. The worksheet automatically recalculated the link margin when the range (distance between the satellite and the ground station) was changed, so that the link margin could be verified at many different ranges. At 43378 km, or 6.801 Re, the link margin calculated in the spreadsheet was 18.04 dB and the link margin calculated by the communications model was 18.03 dB. This is an error of 0.055%.

The Doppler calculations were compared to calculations generated by a program called Satellite Tool Kit (STK). The STK Doppler shift output is similar to the Doppler shift output from the Communications Model. The main difference occurs because the STK calculations were performed based on a different expected launch date. The high-peak Doppler shift value calculated by the STK simulation was 171 kHz while the high-peak value calculated by the Communications Model simulation for the Canberra Ground Station was 167 kHz; this is a 2.34% error. For the low-peak value, the STK simulation calculated it to be –147 kHz and the Communications Model (Canberra Ground Station) calculated it to be –148 kHz, an error of 0.68%. The Canberra Ground Station calculations were chosen for these comparisons because the shape of the shift most closely matched that calculated by STK and it is probable that the STK calculations were done for the Canberra Ground Station.

The Doppler section of the Communications Model was also reviewed by John Staren, NASA/GSFC, who verified that the calculations were performed correctly. John has worked with the ST-5 communications system and generated the STK calculations used to verify the accuracy of the Communications Model Doppler calculations.

Communications Block Summary

The communications model is used to calculate the link margin, Doppler shift, and ground station visibility for those times in the orbit when the satellite is scheduled to transmit. An initialization file and the Orbit Propagator model must be run prior to running the communications model to properly set all of the input variables. After the input variables are set, the communications model can be run. An S-Function computes the ground station look angles, while block models are used for the link margin and Doppler shift calculations. When the communications model is finished running, an output file is created showing the times when the satellite can transmit to each of the ground stations, as well as the link margin and the Doppler shift for each of the ground stations.

3.2.4 Orbit Propagator Model

The Orbit Propagator model was originally created by Jim Morrissey (Appendix 2) and uses some basic physics to determine the orbit of ST-5 based on an initial position and velocity. The math that the model uses is based on two-body orbital mechanics (the earth and the satellite).

Newton's Second Law states that the rate of change of momentum is proportional to the forces impressed and is in the same direction as that force. This can be applied to our model in conjunction with Newton's Law of Universal Gravitation. Newton's Law of Universal Gravitation states that any two bodies attract one another with a force proportional to the product of their masses and inversely proportional to the square of the distance between them. Since we are using a two-body model, the only force that is acting on the satellite is the gravitational force from the earth. According to Newton's Second Law, this would cause the satellite to be constantly accelerating toward the center of the earth.

The acceleration due to gravity is easily calculated. As stated in Newton's Law of Universal Gravitation, it is a function of mass and distance. Since the masses of both the earth and the satellite are known, given an initial position, we can calculate the acceleration of the satellite.

Completing the calculation of the acceleration based on the current position requires some vector mathematics. This may seem complicated, but Simulink allows us to graphically represent this mathematical problem as a model, making it much easier to understand. Figure 22 shows the calculation of the acceleration vector of the satellite due to gravity, based on the position vector of the satellite.

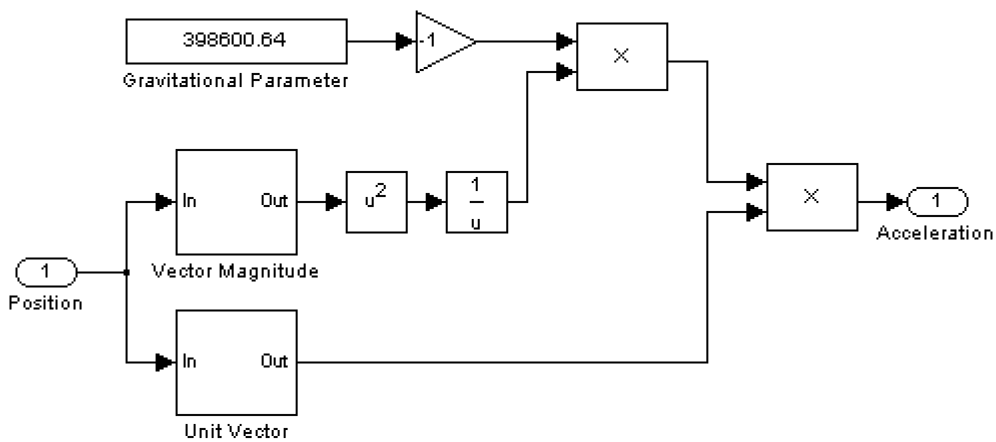


Figure 22: Acceleration Vector Calculation

Once the acceleration is calculated the signal is integrated to find velocity. Once velocity is calculated it is integrated to determine the next position. Figure 23 shows the calculation of the position vector based on the acceleration. The previous velocity and position are used as the initial conditions for these integrations. Now that the next position is known, it can be routed back into the acceleration calculator to restart the process. If this process is continued for a sufficient amount of time, an entire orbit will be propagated.

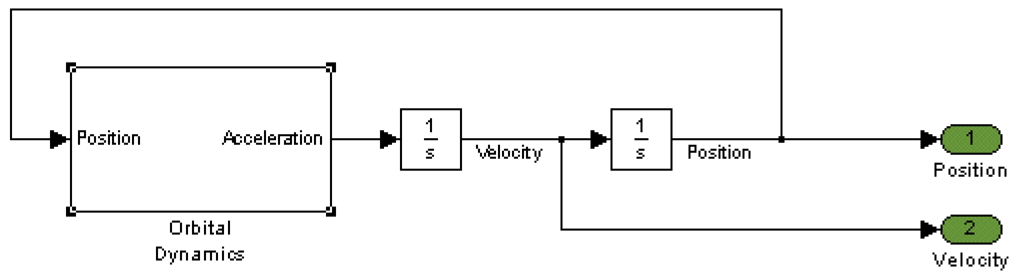


Figure 23: Position Vector Calculation

In order to use the values calculated by the orbit propagator in other models, the values must be saved into the Matlab Workspace. The values are saved into the Workspace by a separate block, which was created specifically for this purpose. The values are saved into a matrix with four columns. The first column is the time, followed by the X, Y and Z ECI coordinates of the satellite at that time (Figure 24).

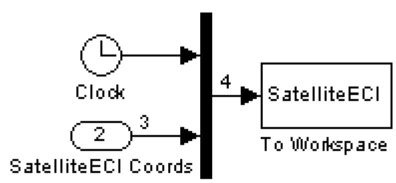


Figure 24: Satellite Coordinates to Workspace

With the ECI coordinate values in the Workspace; they can be used to create a plot of the expected orbit. We can visually verify the coordinates by plotting the X, Y and Z coordinates in a three dimensional axis. Figure 25 is a graphical representation of the satellite's three-dimensional ECI coordinates as it orbits the earth.

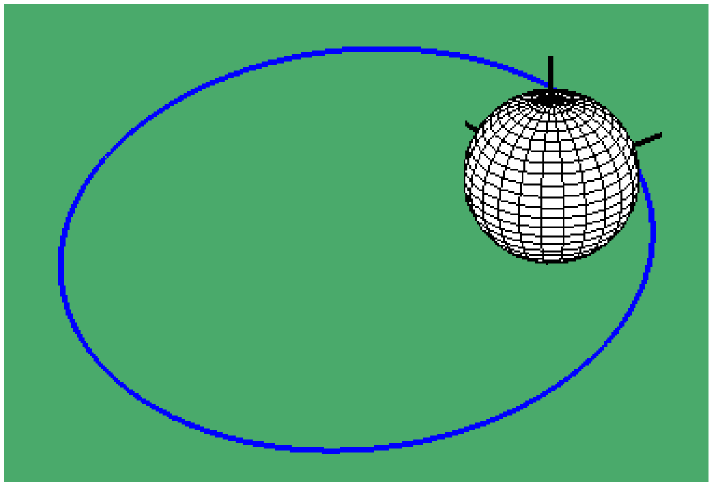


Figure 25: Satellite Path as it Orbits the Earth

Eclipse and Illumination Determination

As the satellite orbits the earth, it moves in and out of the earth's shadow. Because the satellite relies heavily on the sun for power, it was very important to the mission that we determine whether the sun illuminates the satellite or not. The calculation that determines this is based on two angles. The first angle is between the satellite position vector and the sun to satellite vector (Figure 26, Θ_1). The angle between the position vector of the satellite and the line tangent to earth that connects with the satellite is the second angle (Figure 26, Θ_2). When Θ_1 is less than Θ_2, the earth occults the satellite and the satellite must run on its battery.

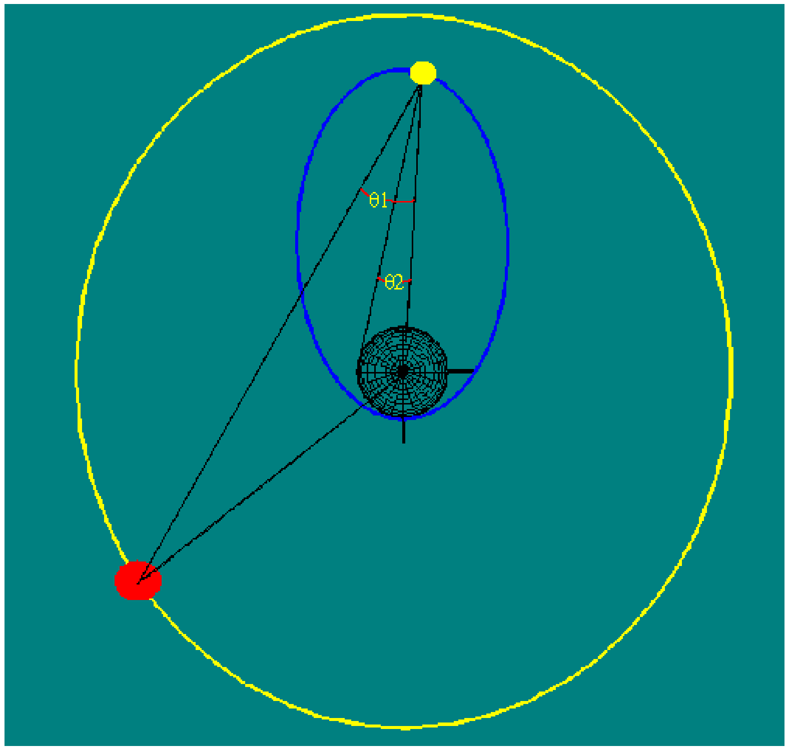


Figure 26: Orbit with Sun Position

In order to determine the illumination status of the satellite we needed the ECI coordinates of both the satellite and the sun. The ECI coordinates of the satellites were already being calculated, but the addition of a sun propagator was necessary to determine the ECI coordinate of the sun.

The sun propagator (Figure 27) we included in our model was primarily controlled by an S-function. It requires a Julian date as an input and will provide the sun ECI coordinate as outputs. With the information provided by both the Orbit propagator and the sun propagator we began to calculate the necessary angles.

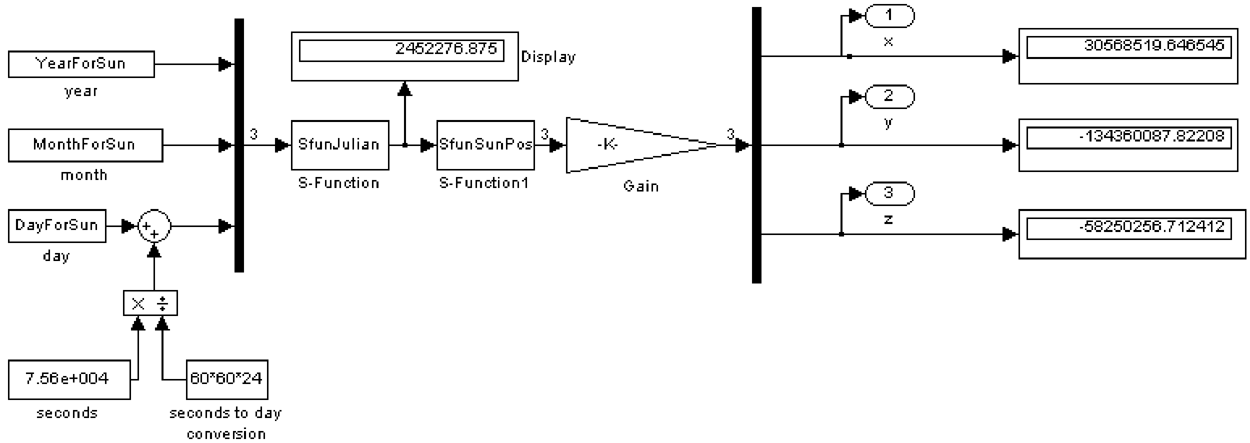


Figure 27: Sun Propagator System

Calculating the necessary angles required the use of vector math and trigonometry. Since we know both the sun vector and the satellite vector, we can determine the angle between the two by using the definition of a dot product (Figure 28).

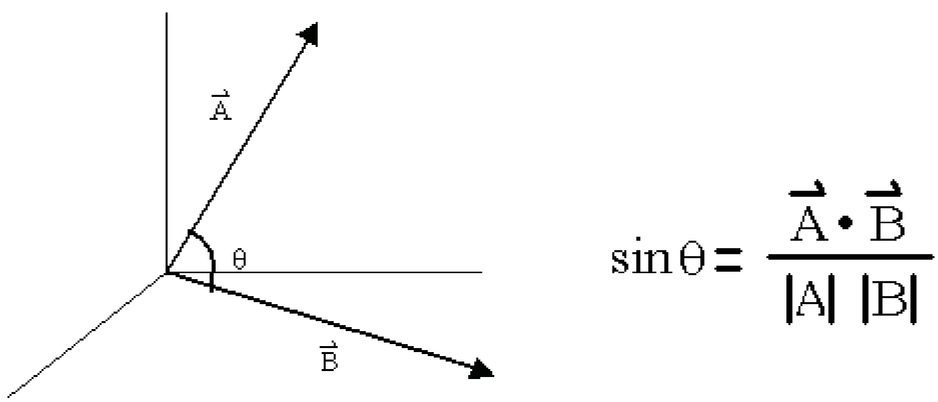


Figure 28: Definition of a Dot Product

The next angle we need to determine is between the satellite vector and the line from the satellite that is tangent to the earth. This is done using trigonometry functions Figure 29.

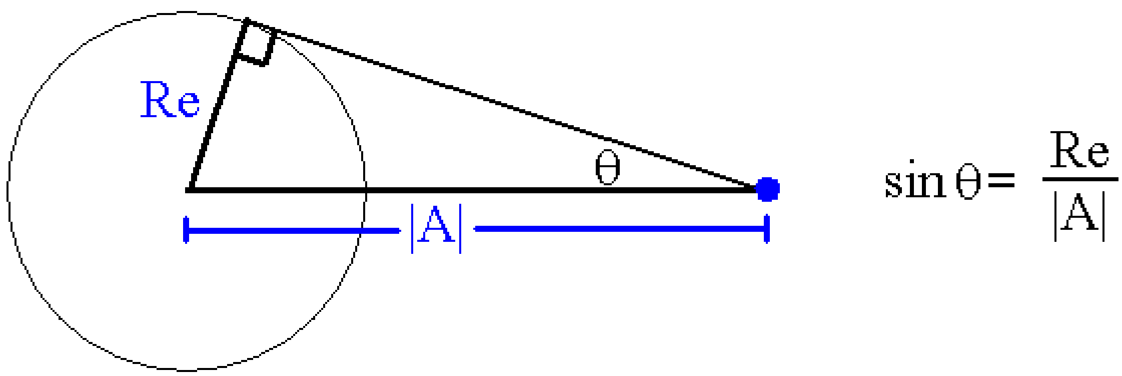


Figure 29: Calculating Angles using Trig Functions

The dot product (Figure 28) and trigonometric math (Figure 29) calculations were completed using a Simulink model designed to solve for the necessary angles. The block receives the ECI coordinates of the satellite and the sun and outputs the two desired angles (Figure 30).

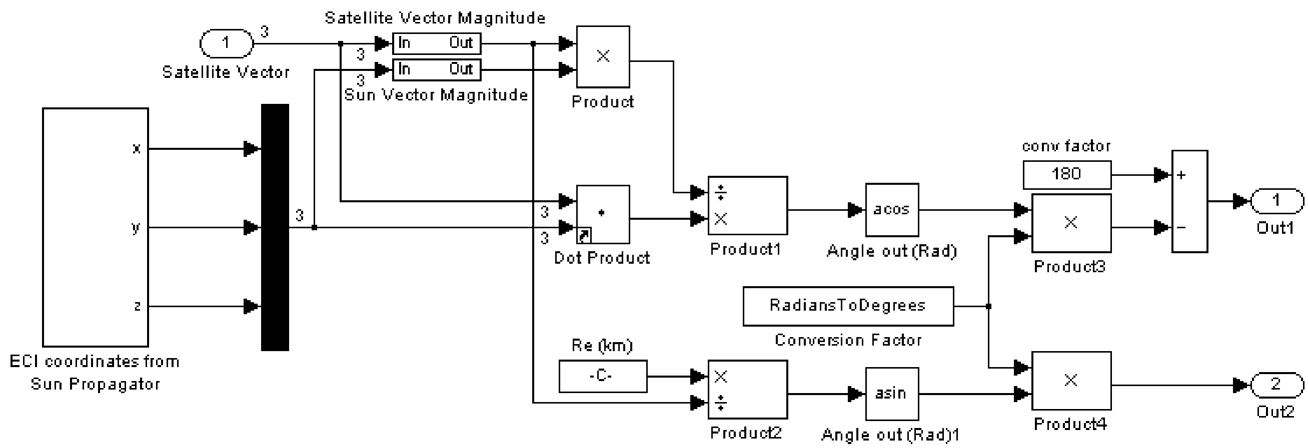


Figure 30: Angle Calculation using Simulink

The angles were then sent into a logic block that determined which of the angles were larger. The block would then output a one if the satellite was illuminated or a zero if the satellite were in eclipse. The final step was to record this information into the Matlab Workspace so the other Simulink models could access the data that was determined in this model (Figure 31).

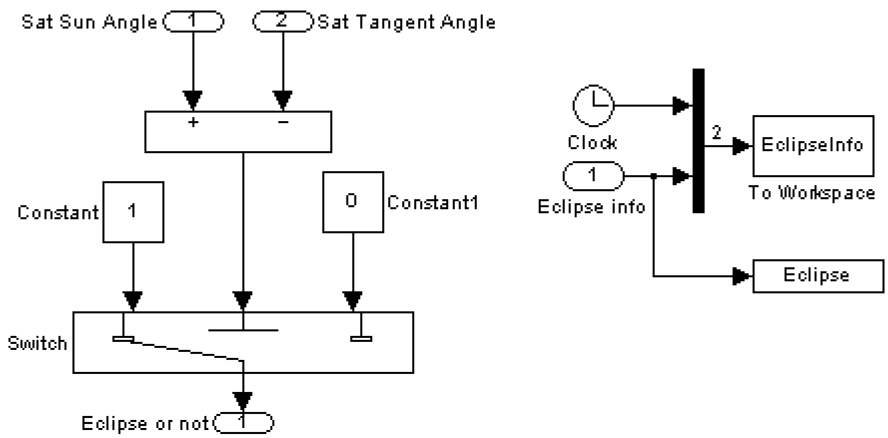


Figure 31: Angle Logic sent to Workspace

3.2.5 Incorporating Operational Scenarios

At the time of launch, a flight plan for ST-5 will be mapped out that will indicate at any given time several parameters, including: where the satellites should be, what the satellites should be doing, and what the health status of each satellite should be. The flight plan will also provide information that is critical to the mission, such as the range of the satellites, what ground stations are available for communication, the Doppler shift, and the power of each satellites' transmissions.

The flight plan will be the result of extensive planning and simulating of the mission. Effort of the ST-5 mission team had already gone into creating several operational scenarios that will be used in the flight plan.

The operational scenarios represent routines that the satellites will perform in orbit and are composed of actions such as initial sun acquisition, maneuvering, activation and use of the variable emissive coating, and the magnetometer activation and use.

In order to test the operational scenarios, they were incorporated into the ST-5 model simulation. As each scenario is run, the simulation represents each action that is taking place. For example, there are two systems that are affected when the magnetometer is activated: the power system because of the load addition, and the data recorder because the recorder data rate will change. In order to account for the activation of the magnetometer, adjustments are made on the load on the power system and the data collection rate. Once the actions in all routines are accounted for in the simulation, each operational scenario can be run to make sure that it will not cause the satellites to fail.

The behavioral model was designed to ensure that the power margin does not drop below a certain level and that the data recorder does not overflow. Since the routines performed in each operational scenario are already determined, the main factor that is determined by the model is how certain environmental changes affect the outcomes of the scenarios. The basic environmental change that takes place throughout the orbits is whether the satellite is eclipsed or not. The purpose of the behavioral model is to model where a satellite will be at a certain time, determine if it will be in eclipse, and determine how the eclipse will affect the systems in the model. The model takes these environmental attributes, compares them with the desired operational scenarios and determines if the scenarios are valid. If the scenarios are not valid, the model indicates why they are not valid, allowing for modification of the scenario(s).

The operational scenarios were easily incorporated into the ST-5 model once it was functioning properly. Simulink allows any input variable to be a time dependent array of values, the time dependent array allows the inputs to be setup so they can hold any desired value at any given time in the simulation. Also, since the inputs can be changed using the time dependent arrays, incorporating the operational scenarios into the model was a matter of setting up the arrays to represent the desired operational scenario.

Our model is meant to emulate the behavior of the ST-5 satellites as they carry out their expected operations. The model is essentially controlled by a series of switches and initial conditions. The switches and initial conditions are defined in Matlab files. Each operational scenario file sets the initial conditions and control arrays (switches) to make the model behave as the satellite is expected to behave.

Simulink stores arrays of numbers in an area called the Workspace. In the Workspace, variables are stored and accessed according to their name, and can contain a single number or a matrix of any dimension.

Example:

Command window line: SwitchEvents = [1 3 1 1 1 0 1 1 1 9 1];

Creates: variable in Workspace called 'SwitchEvents', which refers to a matrix of size 1 X 11.

Once a variable is stored in the Workspace, it can be accessed by a Simulink simulation or by the Command window. If the variable is a matrix, it can be accessed as an entire matrix, a partial matrix, or as a single element of the matrix.

Examples:

Command window line: SwitchEvents = [1 3 1 1 1 0 1 1 1 9 1];

EntireMatrix = SwitchEvents;

Creates: variable in Workspace called 'EntireMatrix', which refers to a matrix of size 1 X 11. 'EntireMatrix' values would be the same as SwitchEvents.

Command window line: SwitchEvents = [1 3 1 1 1 0 1 1 1 9 1];

PartialMatrix = SwitchEvents(1,2:10);

Creates: variable in Workspace called 'PartialMatrix', which refers to a matrix of size 1 X 9. The elements in 'PartialMatrix' would reflect columns 2 through 10 of the first row of 'SwitchEvents'. 'PartialMatrix' = [3 1 1 1 0 1 1 1 9]

Command window line: SwitchEvents = [1 3 1 1 1 0 1 1 1 9 1];

SingleElement = SwitchEvents(1,2);

Creates: variable in Workspace called 'SingleElement', which refers to a matrix of size 1 X 1. The number in 'SingleElement 'would reflect the number located in row 1, column 2 of 'SwitchEvents'. 'SingleElement' = [3]

As demonstrated above, creating a workspace full of variables can become very tedious, especially when the model simulation requires approximately 152 variables of varying sizes in the workspace before it can run properly. In order to prevent the tedious process of entering each initial value, we created Matlab files to initialize the Workspace. We created five files to simulate the five operational scenarios we were given.

Each scenario has its own set of variables that need to be generated. The most important changes are made in the 'SwitchEvents' array, which controls when each subsystem is activated.

Other important values:

InitBOLEOL	- indicates the age of the satellite and modifies the behavior accordingly.
InitialSOC	- indicates the initial state of charge of the battery as a percentage
nOrbits	- states how many orbits will be carried out by the simulation
LoadMargin	- adds a margin of error to the load required by each subsystem

Sample operational scenarios are shown in Appendix 6.

3.3 Summary

The original ST-5 model was updated in several ways in order to provide a more accurate simulation of the behavior of the spacecraft. In particular, the model was updated to include functional models of the communications subsystem, the data recorder, and an orbit propagator. The voltage regulator model was also validated by developing a pulse-width modulating regulator and comparing it to the given regulator model. The new models were incorporated into the model by altering the set up files and the power model provided by Scott Starin.

The final ST-5 model is intended to simulate the behavior of the spacecraft with respect to communication information, status of the data recorder, and health status of the power model while running given operational scenarios. The orbit propagator model was updated to provide important information necessary to calculate several factors that were dependant upon position, including information needed for the communications subsystem. The communications model was designed to calculate Julian Date, Greenwich Sidereal Time, ground station sight, link margin, and Doppler Shift. The data recorder model was also developed to track the amount of data recorded, the amount of data needed to be transferred, and alert the user of overflow. Finally, operational scenarios were incorporated into the model in order to run a behavioral simulation of the spacecraft under given conditions.

4. Results

4.1 Introduction

The behavioral model that was created was used to run operational scenarios that were generated by NASA as projected flight scenarios. The behavioral model consists of four sub-models: an orbit propagator model, a power subsystem model, a data recorder model, and a communication subsystem model. The models generate several outputs to the workspace based on the scenario that was run. The scenarios were run with the developed model to generate a series of plots and arrays that determine the status of the power subsystem, the communications subsystem, and the data recorder, along with generating satellite and ground station ECI coordinates and eclipse information. The generated outputs of the model allow the user to determine any areas in the orbit that may contain failures in any of the modeled subsystems.

The power subsystem model was altered and tested in order to examine the effects of a pulse-width modulator modeled in the voltage regulator. This model was not used in the final behavioral model due to simulation restraints, but was used to compare results of the two different regulator models.

The results of the behavioral model were validated in several ways. The outputs of the communications model were verified based on known values that were verified visually. The results from the data recorder were verified using calculated values. The model was thoroughly tested against known values and calculated values to ensure that it accurately simulated the subsystems of the spacecraft. The model of ST-5 is very dynamic and can simulate nearly any desired scenario of events including solar array failures, low initial state of charge of the battery, and more based on the variable inputs.

4.2 Voltage Regulator

A realistic model of the pulse-width modulator used in the voltage regulator section of the Power Subsystem model was created and implemented. The model including the

PWM rendered results that very closely mimicked the pre-existing model of the voltage regulator. The new model of the voltage regulator was not used in the final behavioral model because of the increased factor of accuracy required to successfully pulse-width modulate using a 100 kHz triangle wave. The sampling time required to run a simulation of the pulse-width modulation was much too short for the overall accuracy required for the rest of the behavioral model.

The results generated by the PWM voltage regulator model helped to verify the original model. The voltage regulator model created by Scott Starin uses an average feedback loop for shunted power and actual solar array current. We were unsure of the validity of this method, and therefore created a realistic representation of the pulse-width modulation that occurs in the physical voltage regulator circuitry of the Power System Electronics (PSE) (Figure 32). This circuitry serves as the voltage regulating section of the PSE, containing a comparator that compares difference between the bus voltage and the maximum voltage (V_{REG}) to a generated 100 kHz triangle wave.

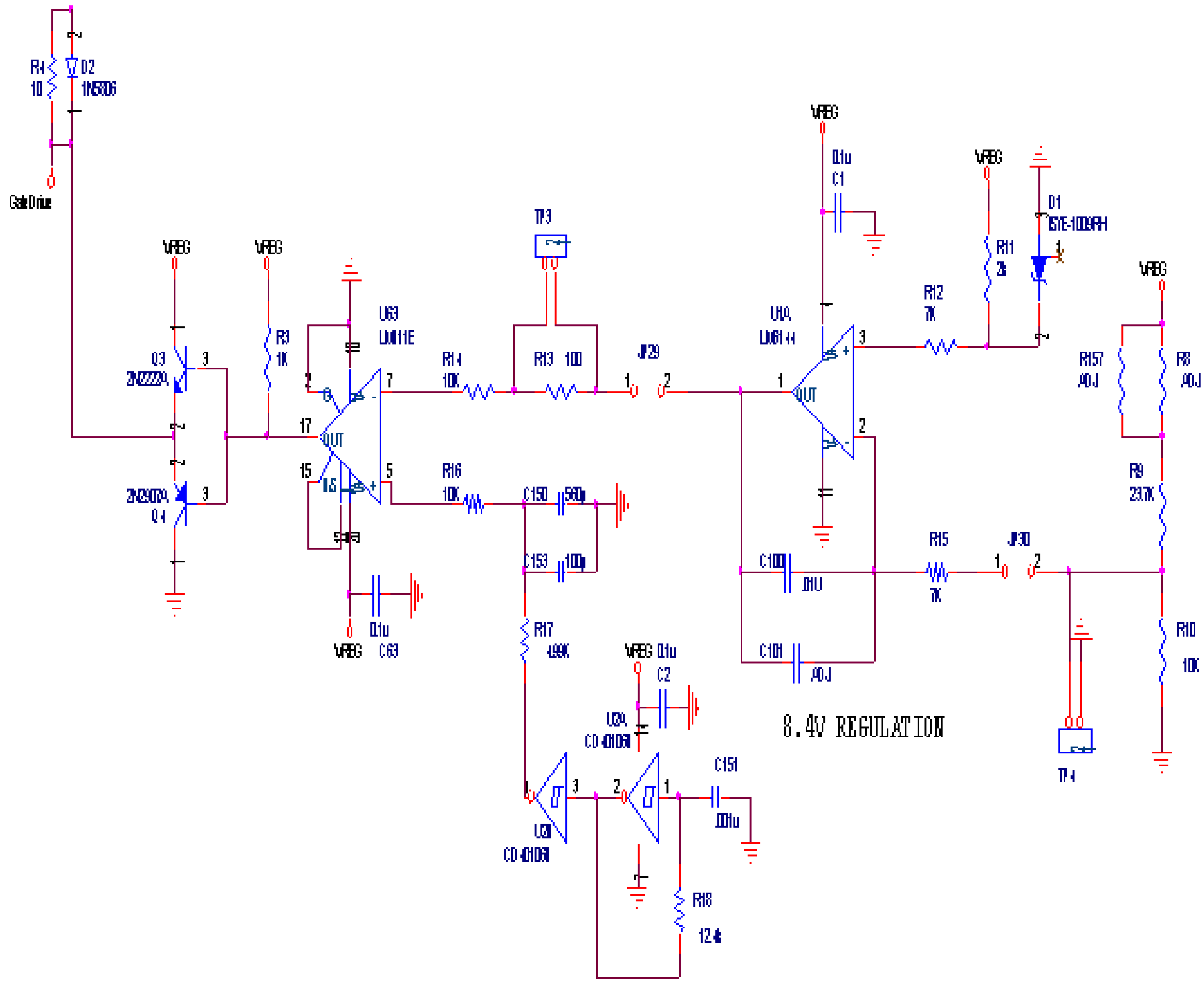


Figure 32: Voltage Regulator Circuitry

Simulations of each regulator model (PWM regulator and original regulator) were run and compared in order to verify the functionality of each. The pulse-width modulator (simulation in Figure 33) compares the integrated error between the bus voltage and the maximum voltage to a sine wave. Figure 33 depicts a simulation of the developed PWM model. The yellow line is the integrated error signal, the pink wave is the reference sign wave, and the blue signal is the pulse-width modulated on/off signal. The figure shows that when the integrated error signal is greater than the reference signal, the power is being shunted (logic high).

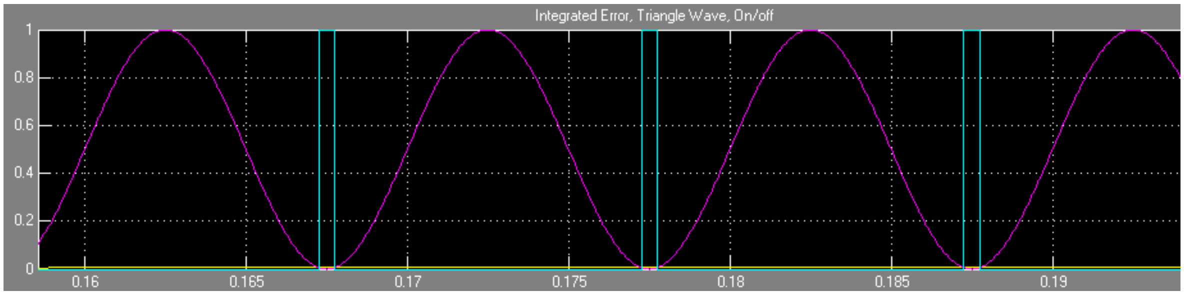


Figure 33: PWM Simulation

Figure 34 shows the effects of the PWM on the bus voltage. This simulation is pulse-width modulating at 100 Hz. The simulation shows a small variation on the bus voltage (1mV) as a function of the on/off shunting logic seen on the third scope. As expected, the bus voltage decreases when the shunting logic is high. There is minimal variation on the shunted and actual current on the second scope in Figure 34.

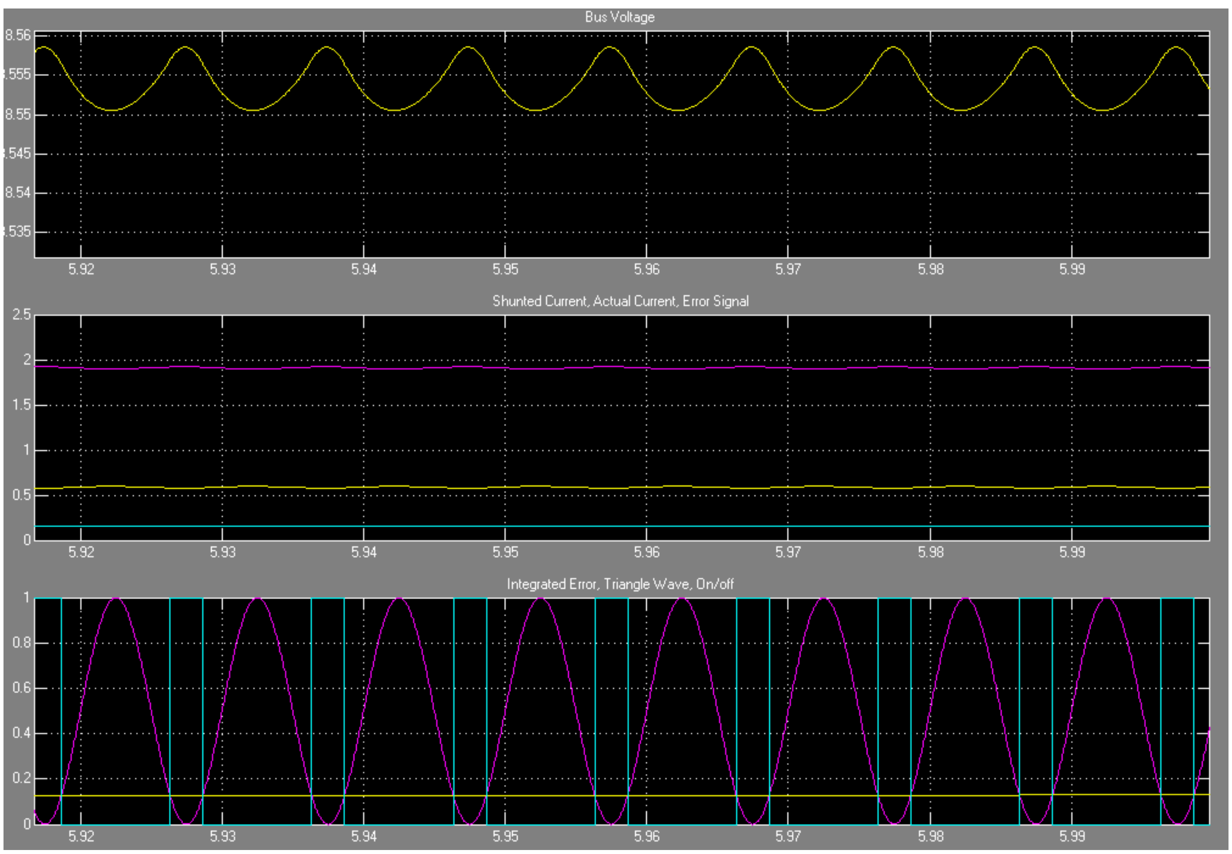


Figure 34: PWM Bus Voltage

Both models rendered very similar results and it was determined that the accuracy of the pulse-width modulation model was not needed to accurately simulate the ST-5 power subsystem. Simulations of the two different models are pictured below in Figure 35 and Figure 36. Figure 35 is a simulation of the voltage regulator with pulse-width modulation. This simulation is pulse-width modulating at 100 Hz and filtering the output at 10 Hz using a Butterworth Filter. The values used were the highest frequencies that Simulink could effectively simulate due to the small step size required for the filter (50 μs). Because of hardware constraints, this simulation could only run for 40 s. The time that the pulse-width modulation regulator took to run was significantly longer (approximately 25 min. to simulate 40 s of an orbit) compared to the original voltage regulator simulation (under 1s to simulate 40 s of an orbit) located in Figure 36. As seen below, both regulator models generate similar results, with minimal variation seen due to the limited sampling abilities of Simulink for the pulse-width modulation.

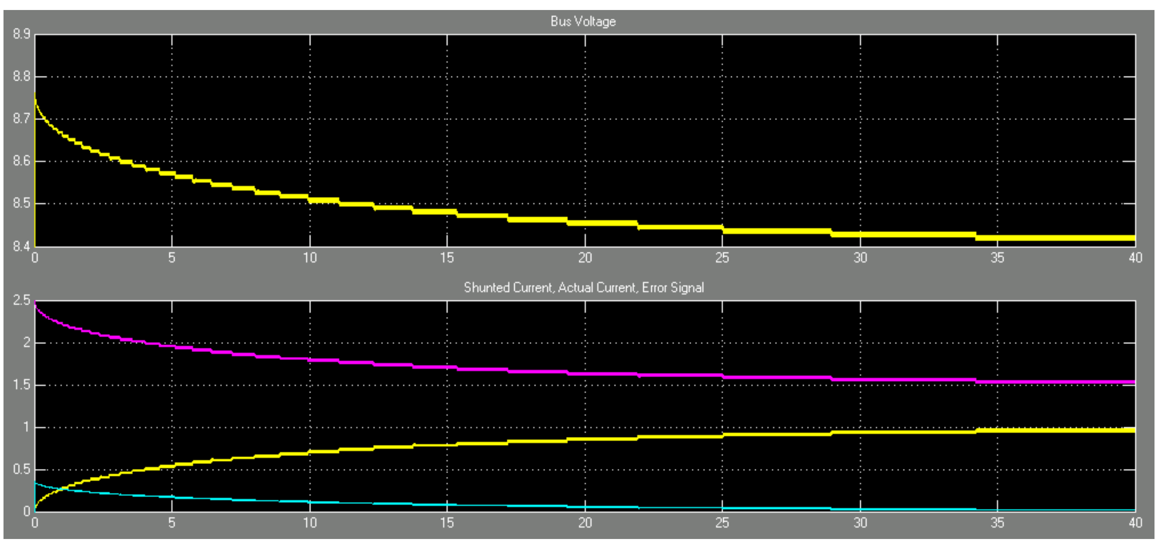


Figure 35: PWM Model Results

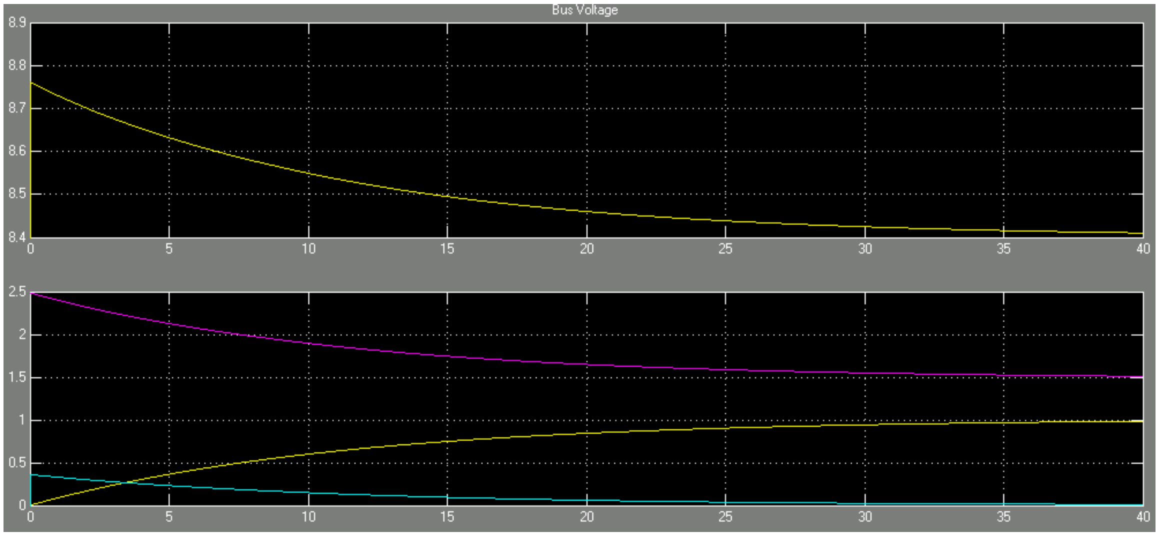


Figure 36: Original Regulator Results

The voltage regulators were also tested at different initial states of charge (SoC) and rendered similar results when the initial state of charge (SoC) of the battery was not 100%.

The original model developed by Scott Starin was implemented into the behavioral model because of increased speed and the capability to handle a much larger sampling time, while maintaining accuracy. The model that implements an average method of shunting power, rather than pulse-width modulation, renders results that are accurate for this model at a reasonable sampling time and running time compatible with the overall behavioral model. It was determined that the pulse-width modulation regulator was not a practical method of modeling the voltage regulator.

4.3 Behavioral Model Results

The model generates several sets of results in the form of arrays to the workspace. Some of the arrays generated do not help the user understand the status of any of the subsystem, but are generated in order for the model to operate correctly. There are, however, several important arrays that are generated as outputs. In order to observe the status of the output arrays we were concerned with, we wrote a Matlab script file to generate a series of plots.

The plot script file generates graphical representations of several variables for each sub-model of the behavioral model. The generated plots serve show critical information such as ground station visibility, transmission times, and orbit information. The plot script file is used and run separately with each operational scenario to graph the status of the satellites' different subsystems under the conditions of the particular scenario.

We were given 5 operational scenarios that were updated to interact with our behavioral model. Each scenario depicts a series of different events that make it unique. When the plots script file was run with each, different outputs were plotted, which allowed us to see the effects of the different scenarios on the satellites. Operational Scenario 5 (Appendix 7) was used to generate sample plots from the script file. Scenario 5 models the satellite while it is activating and using the magnetometer to collect science data. The magnetometer affects the load on the power model and also the recording rates of the data recorder. The effects can be noted in the plots that were generated from our script file.

Figure 37 shows the eclipse regions of the satellite. Our model has two methods of generating eclipse data. The first plot in the figure depicts the eclipse data that is generated by the orbit propagator model. The second plot in figure shows the eclipse data that is generated in the operational scenarios as the variable "IsLit2".

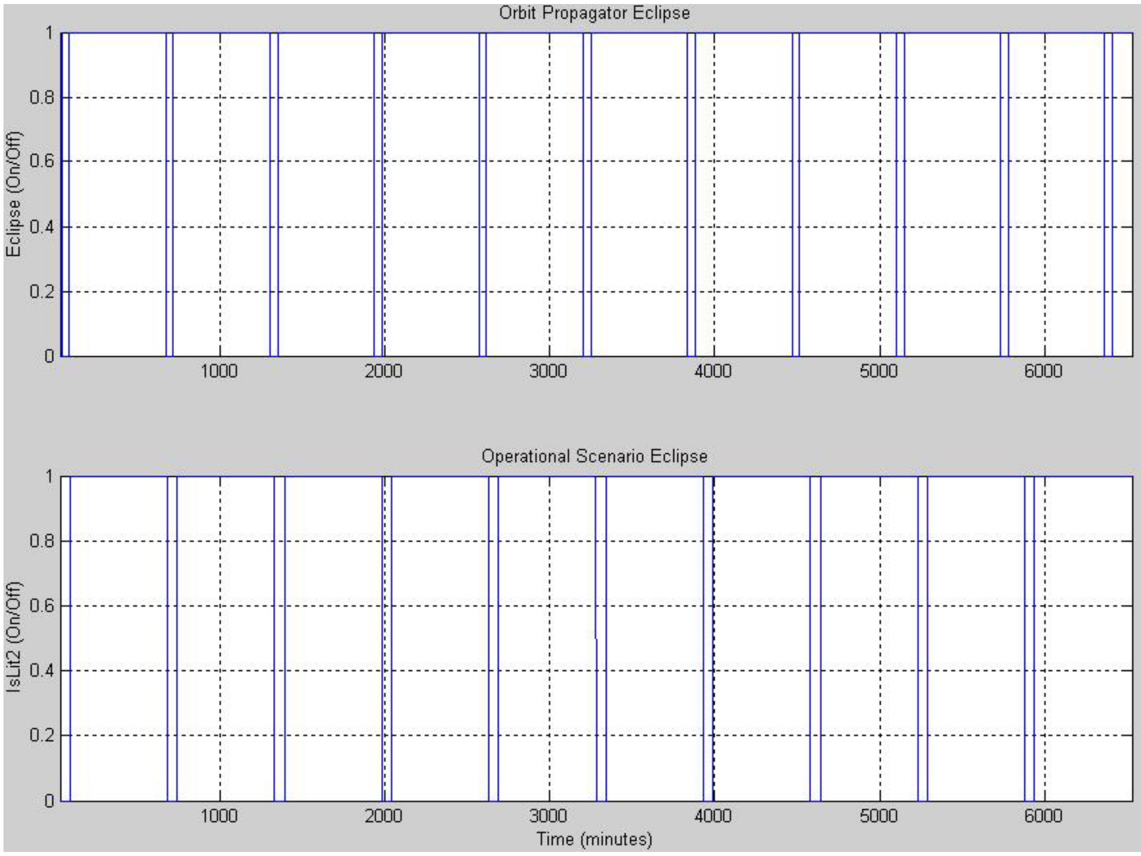


Figure 37: Eclipse Periods

Figure 38 generates four different plots of outputs. The outputs plotted in Figure 38 are: eclipse regions (from operational scenarios), transmission periods, ground station visibility, and the state of charge (SoC) of the battery. The plot of eclipse periods in Figure 38 shows the times in which the satellite is eclipsed for case 5 as generated in the scenario. The second plot in figure shows the transmission periods of the case. The scenario assumes one transmit period per orbit (as expected), but can easily be altered in other scenarios to simulate the spacecraft to miss a transmission. The visibility of the three ground stations (Canberra, Goldstone, and Madrid) is shown in the third plot of Figure 38. The visibility of the ground stations is generated from a combination of the orbit propagator and the communications block. The last plot in Figure 38 is a plot of the spacecraft battery's state of charge (SoC). For case 5 the battery SoC does not drop below 85%, which is within the limits of the minimum depth of discharge set for the mission. The SoC is shown to drop when the satellite is transmitting and during eclipse periods. The transmit periods occur directly after the spacecraft comes out of eclipse, as set by the operational scenario.

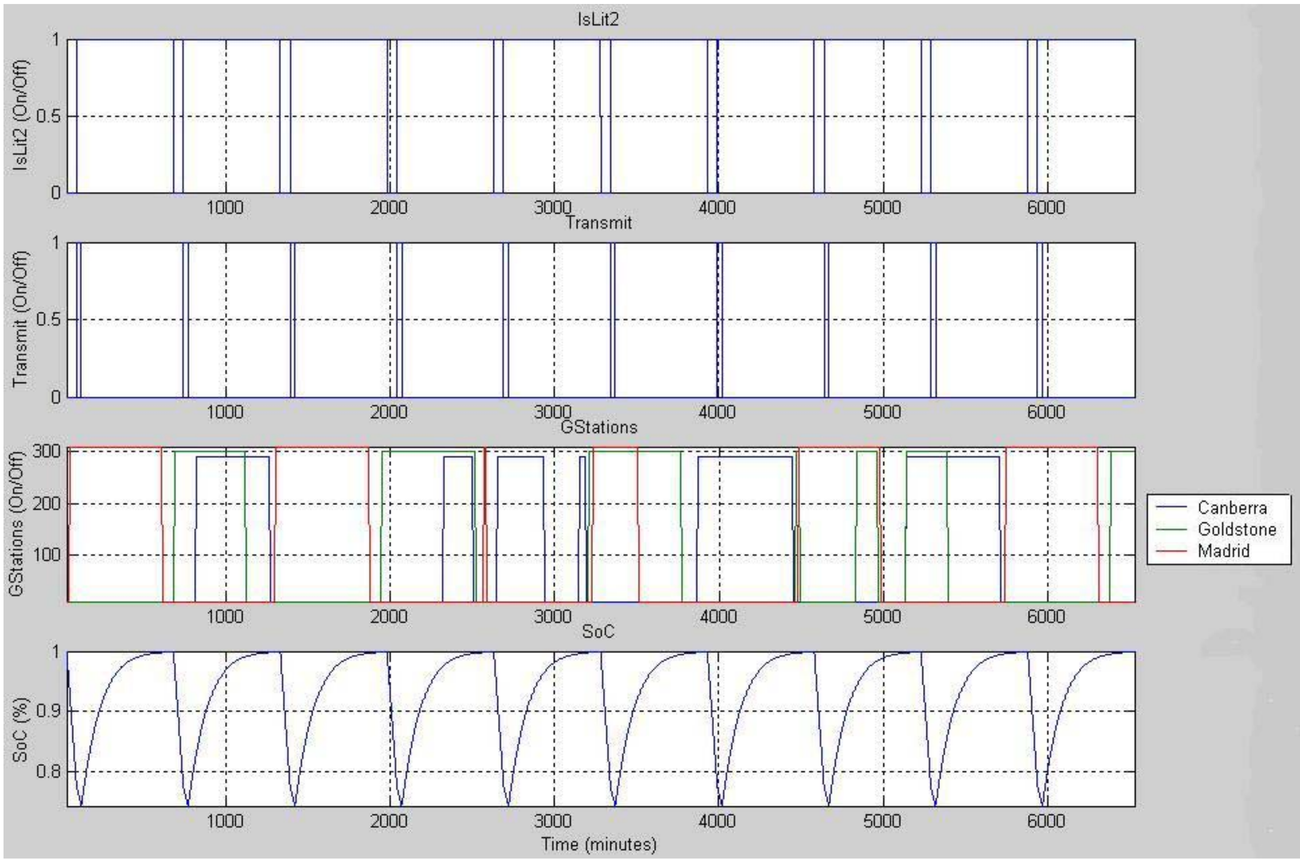


Figure 38: Eclipse, Transmit, Ground Stations, SoC

Figure 39 shows the plots of the ground station visibility (Figure 38), Doppler shift, and link margin. Knowing the Doppler shift allows the ground station adjust its transmission frequency accordingly before a transmission occurs. The link margin must be at least 3dB in order to carry out a successful transmit. As seen in Figure 39, the link margin of ST-5 never goes below 18dB in case 5.

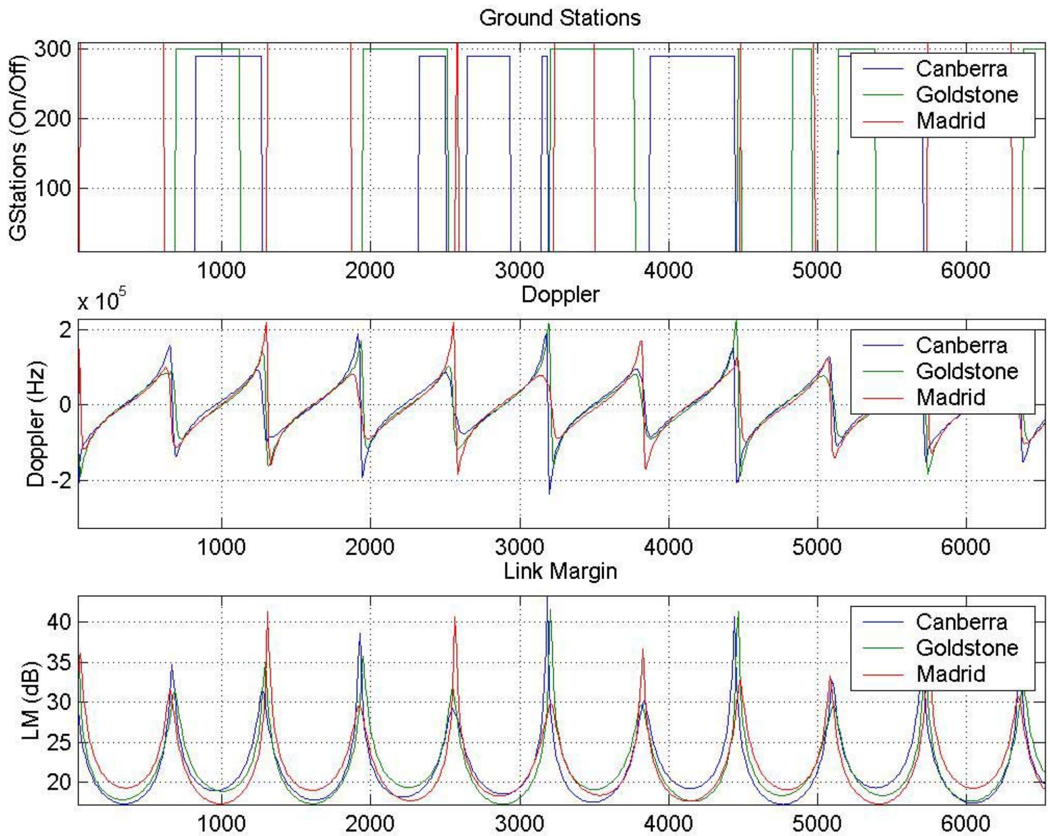


Figure 39: Ground Station Visibility, Doppler Shift, and Link Margin

Figure 40 shows some of the outputs generated by the data recorder including: the telemetry and transmit rates, the data transmitted, and the total data. The first plot in Figure 40 shows the telemetry and transmit rates which. For case 5 the telemetry rate is constant because there is only one filter table used, which is easily altered for other cases. The transmission rate of the model is a constant rate, shown in Figure 40. The second plot in Figure 40 is a plot of the data to be transmitted, which shows the spikes of data that is transmitted at each transmission. Case 5 does not result in any problems transmitting all of the data necessary to transmit at each pass, shown in Figure 40. The last plot in Figure 40 is a plot of the total data that reacts to transmissions by releasing the amount of data stored in the partition of memory reserved for a previous orbit of data (also plotted in Figure 41).

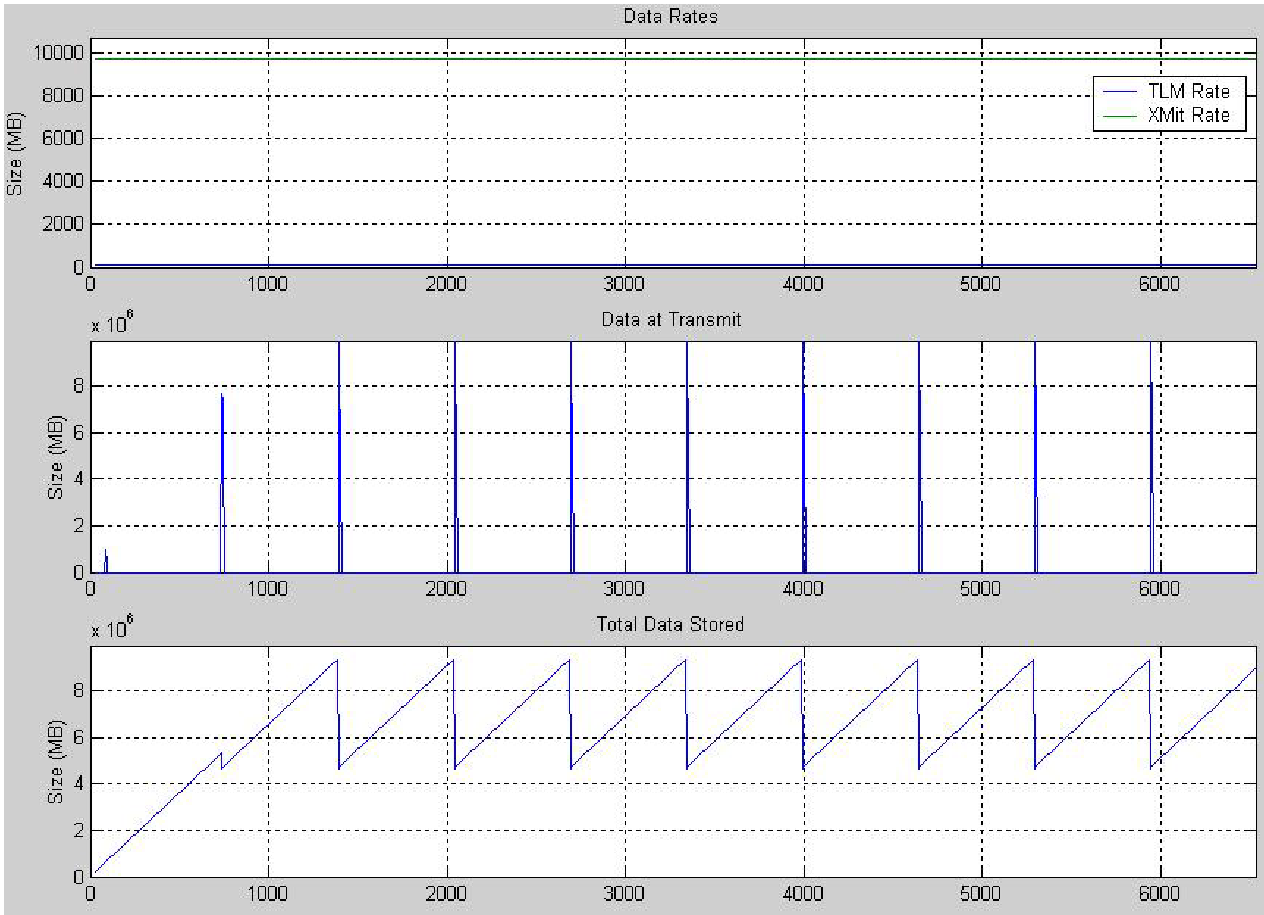


Figure 40: Data Rates and Transmissions

Figure 41 shows some data recorder plots. The transmit periods are plotted indicating the points in the scenario in which the satellite is in transmit mode and the high-powered amplifier is on. The second plot in Figure 41 shows both the current data and previous data recorded. The current data always consists of the most recent orbit collection and the previous data contains the previous orbit collection. At a transmission the previous orbit collection is released (deleted) and the current data is moved into the previous data partition (Figure 41). The third plot in Figure 41 is the total data stored (also seen in Figure 40), which is the sum of the current data and the previous data. The last plot in Figure 41 is a plot of the data transmitted (also seen in Figure 40). This plot shows the data that is transmitted at each transmission time. This information is important because it allowed for us to determine whether all of the data in the recorder at the time of transmission was downlinked.

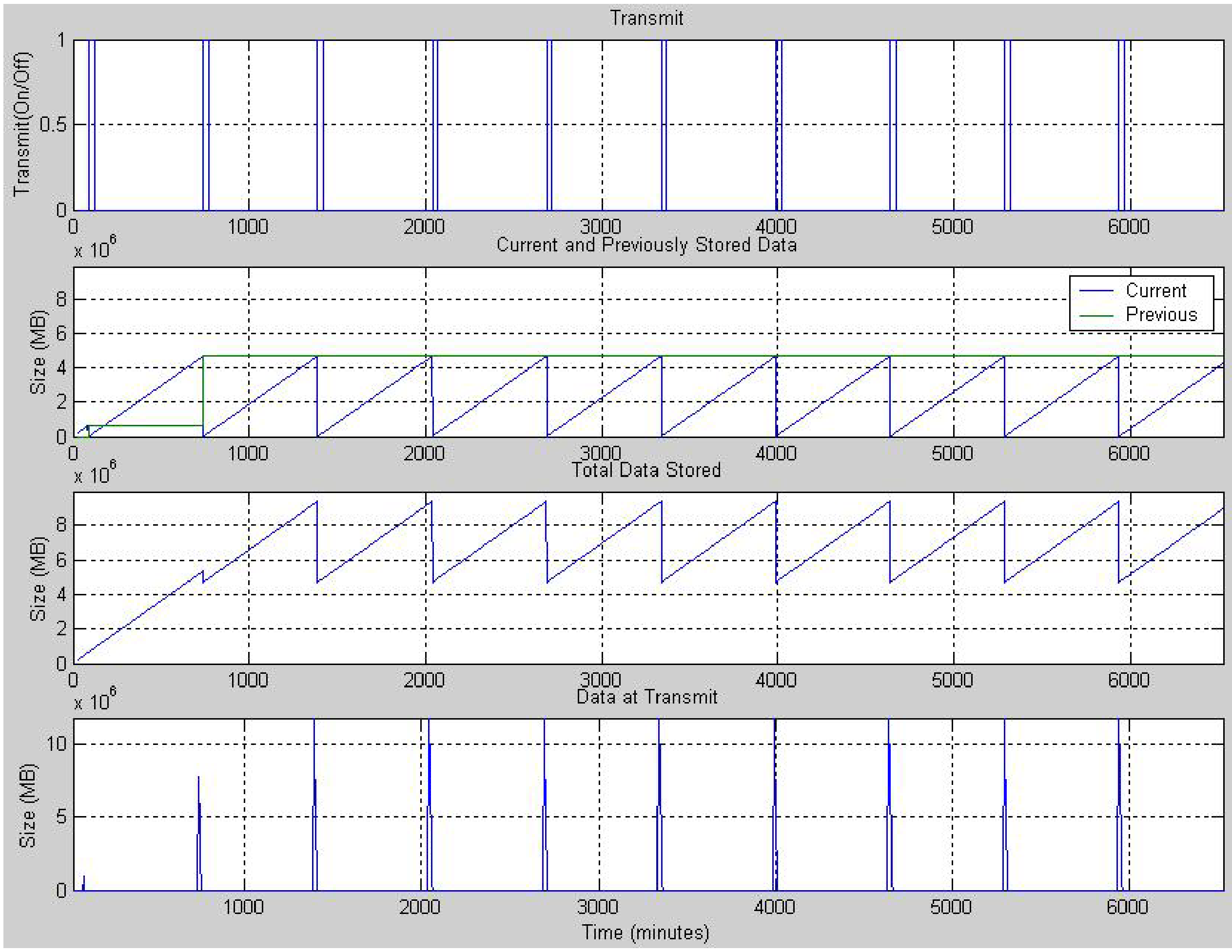


Figure 41: Transmission and Data Collection

Figure 42 is plot of the load dissipations. This plot is very informative because it allowed us to see what loads were on over the time of the scenario. Figure 42 plots all of the loads on the power system:

transmit and receive mode (TR) the spacecraft has a load drawn from it due to the transmit and receive mode that varies according to transmission and activation of the high-powered amplifier

high-powered amplifier (HPA) the HPA is only activated during transmissions and is the largest load on the power subsystem

pressure transducer (PTr) the pressure transducer is activated when the thrusters are activated and only draws a small amount of power.

magnetometer (Mag) the magnetometer is activated in SwitchEvents (Case5 – Appendix 7)

magnetometer boom actuator (Mba) the magnetometer boom actuator is only activated when the boom is being extended (Case 1)

command and data handling (CDH) the command and data handling of the spacecraft is one of the subsystems of the spacecraft that is always on, even during eclipse periods.

miniature spinning sun sensor (Sun) the sun sensor is also an element on the spacecraft that is always activated, including eclipse periods (Figure 37)

thrusters (Thr) the thrusters are set by the operational scenario to turn on once an orbit

variable emittance coatings (VE1 and VE2) the VECs are activated under the operational scenarios in Switch Events

power system electronics (PSE) the PSE are always activated and serve to regulate the voltage to the battery

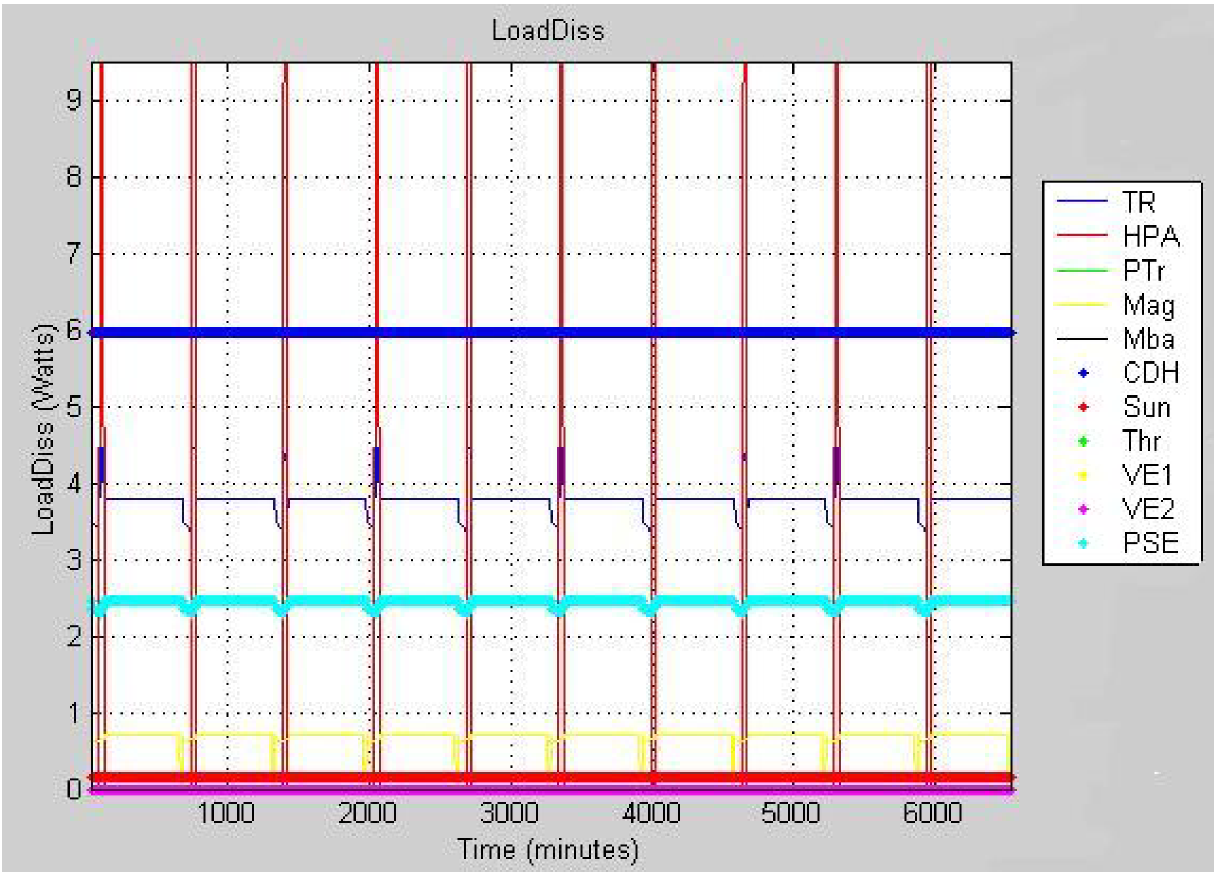


Figure 42: Load Dissipations

Figure 43 is a figure of the plots the satellite orbit (large blue orbit), the sun orbit (large yellow orbit), and the ECI coordinates of the three ground stations – Canberra (yellow), Goldstone (red), and Madrid (blue). This plot allowed us to see where the ground stations, satellite, and sun were at the end of the orbit, by showing up as a different colored point on each line, for example the sun ended in the position of the red dot located on the lower right-hand section of the orbit in Figure 43. Figure 43 can be rotated in the plot window in order to view different perspectives.

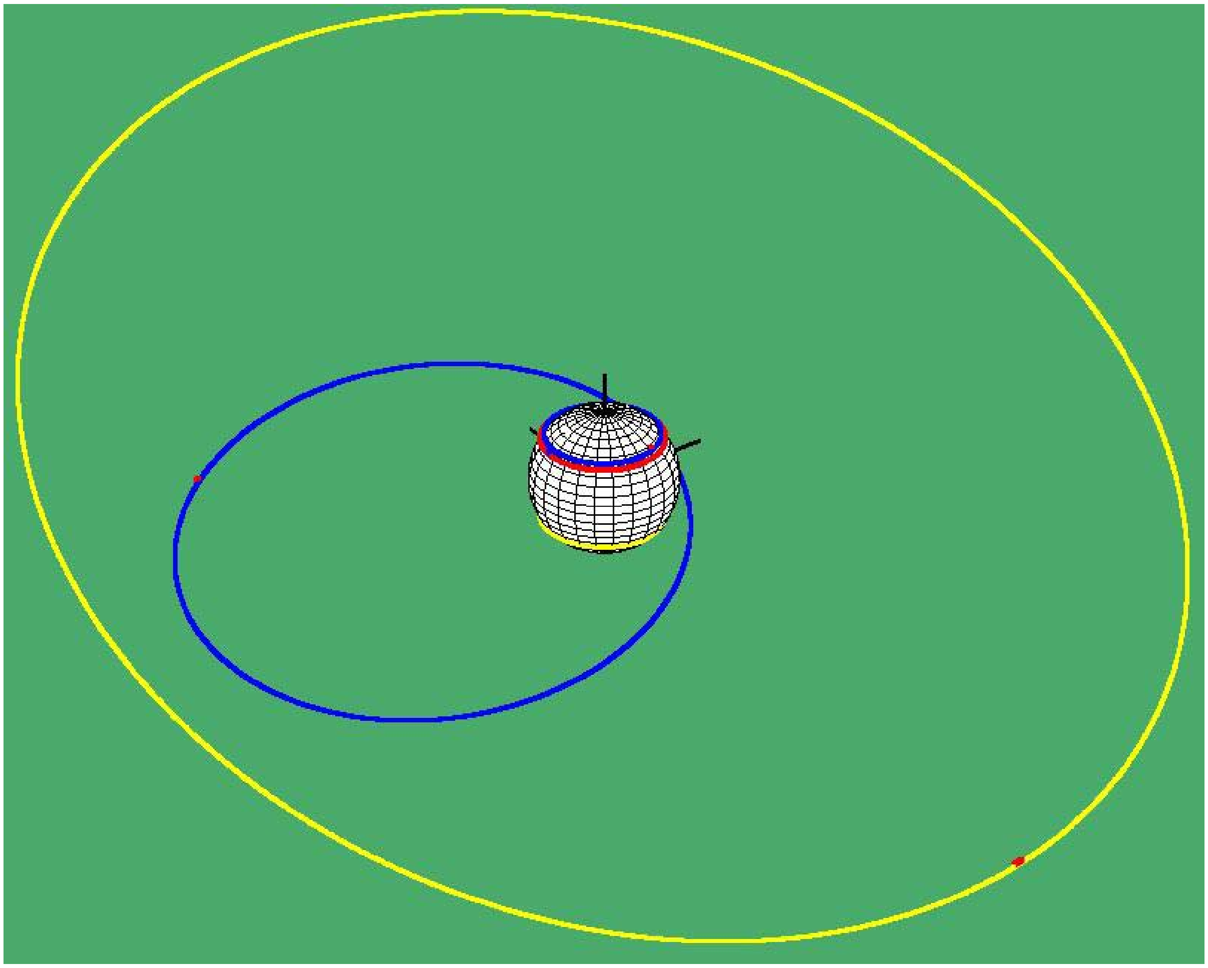


Figure 43: Orbits and ECI Coordinates

Figure 37 through Figure 43 showed the outputs of the behavioral model for operational scenario 5. Each case can be run to generate similar figures that output the characteristics of that particular case. The figures allow the user to view the effects each of the cases' events have on the critical statuses of the spacecraft by simply running the script file "ModelPlots".

4.4 Summary

The behavioral model that was developed generates several outputs as arrays to the workspace based on the four sub-models that were created: the orbit propagator model, the power subsystem model, the data recorder model, and the communication subsystem model. The model was also tested with a pulse-width modulating voltage regulator, but it was determined that this particular regulator model was not functionally adequate to run long length scenarios due to simulation constraints.

Multiple operational scenarios can be created by changing several different variables and by altering the SwithEvents array, which changes the activation of different loads. The results presented represent one particular scenario, but it should be noted that this model is very dynamic and can produce results for any number of scenarios, based on what the user is trying to simulate or test. Assuming the user has a good understanding of the Matlab and Simulink software, the operational scenarios can very easily be altered to produce different events of interest, due to the flexible nature of the behavioral model.

5. Conclusions

5.1 Introduction

In this chapter we describe a dynamic and user-friendly behavioral model of critical systems in the ST-5 spacecraft, created using the Simulink modeling software. The model is flexible and general, while still providing accurate and understandable output for each of the systems modeled, resulting in a powerful mission-planning tool. The model can also be used as a tool to aid in creating operational scenarios and shows great possibility for real time applications, such as operator training.

5.2 Model Capabilities

Through the use of the case files, the ST-5 Simulink model can be used to simulate any operational scenario simply by changing the parameters in the case files. All initial model inputs are accounted for in the case files, from the battery state-of-charge, to the size of the data recorder, to the ground stations' Latitude and Longitude positioning, allowing for easy modification of initial inputs to account for changes or situations that may occur. After a simulation has been run, a script file is used to generate the output graphs from the model. The graphs put together the information from each of the subsystems, providing the combined results from the scenario in an easy-to-understand format. In particular, the output graphs bring together information that could be generated in other ways, such as the ground station visibility and the eclipse information for a particular orbit, and combine that with information that is not generated elsewhere, such as the state of the data recorder, providing a single, powerful, integrated mission-planning tool.

The model can be used not only to simulate planned operational scenarios, but can also be used as a tool to generate new operational scenarios. For example, the orbit propagator can generate position and eclipse information for any orbit. The position and eclipse information can then be directly input into the rest of the model and run with a pre-existing operational scenario to determine the effect of the altered orbit on satellite operation, or the orbital information can be analyzed manually to determine the optimal times for operations and used as the basis for a new operational scenario.

There is also the possibility of using this model for real-time applications. Potential real-time applications include use of the model as an operator-training tool or as a verification tool, feeding it accurate real-time data from the spacecraft, to ensure that the spacecraft are performing as expected. Real-time applications would require that the model be modified to run in real time, which would be facilitated using Simulink's Real-Time Workshop toolbox. The Real-Time Workshop generates C-code from Simulink models to increase the speed of a simulation so that it could be time-stepped to a real clock.

5.3 Suggested Model Modifications

There are a number of modifications to the model that would be possible. The modifications would increase the functionality of the model, making it a more valuable tool.

The communications sub-model currently assumes a 10° minimum elevation requirement for the ground station antennas; however, the actual elevation requirement varies depending on the azimuth. An implementation to allow for the actual antenna elevation requirements, as a function of azimuth, should replace the static elevation requirement used in the *Line of Sight Logic*.

Currently the command to transmit is given in the operational scenario and it is assumed that the transmission will occur. The communications block outputs a graph of ground station visibility based on the look angles and link margin calculated, but does not cross check the ground station availability information with the transmit command given at a particular time. It may be useful to implement logic that would verify that a ground station is available to transmit to, as well as checking other critical parameters (such as the power margin) before assuming that the transmission will occur, or at least generating an error message if a transmission was attempted but failed. Although it is possible at this time to determine whether a transmission was successful from a visual inspection of the output graphs (which show the battery state of charge, the ground station availability, link margin, and Doppler shift), but automated logic would make it simpler.

Another modification that would make the model easier to use would be to integrate all model blocks into one model. This was not done because of the varying time steps required for each model block and the increase in simulation time caused by running the entire model at the smallest required time step. Specifically, the power model requires a much smaller time step than the communications model to achieve the same accuracy, but the communications model is significantly slower and causes a large increase in simulation time when integrated with the power model. When integrated, the simulation does still run at a faster than real-time rate, but it was prohibitively slow for the purposes of this project.

5.4 Summary

The ST-5 behavioral model is a powerful and flexible tool. It can be used as a mission-planning tool, to verify or to create operational scenarios that the satellite will execute. The model is easy to operate and, for someone with a basic Simulink background, easy to modify. The model demonstrates the potential of Simulink, and of behavioral modeling in general, for spacecraft planning applications.

Appendix 1: References

Launius, Roger D., The History of Satellites, *Sputnik and The Dawn of the Space Age*, [Online], Available:
http://inventors.about.com/library/inventors/blsatellite.htm [7 October 2002].

Sarsfield, Liam, 1998, *The Cosmos on a Shoestring: Small Spacecraft for Space and Earth Science*, Rand Publications [Online], Available:
http://www.rand.org/publications/MR/MR864/ [7 October 2002].

Stanton, Stuart, 1999, USAFASAT-B Program, [Online], Available:
http://www.usafa.af.mil/dfas [7 October 2002].

About ST-5, Mission Concept, n.d., [Online], Available:
http://nmp.jpl.nasa.gov/ST-5/about/missionconcept.html [7 October 2002].

Savage, Donald, 1999, ST5 Press Release, *NASA Selects Miniature Spacecraft to Test Space Technology,* [Online], Available:
http://nmp.jpl.nasa.gov/ST-5/press.html [7 October 2002].

Raphael, David, 2001, *CULPRiT Specification,* [Online], Available:
http://st5.gsfc.nasa.gov/ST5-495-149.doc [7 October 2002].

Bate, Roger R., Mueller, Donald D., White, Jerry E., 1971, Fundamentals of Astrodynamics, Dover Publications, New York.

Gavrin, Andrew, 1996, Gravity and Satellite Orbits, *[Online], Available:*
http://webphysics.iupui.edu/gpnew/gp2th4.htm [7 October 2002].

Guidry, Mike, 2002, *The Laws of Planetary Motion,* [Online], Available:
http://csep10.phys.utk.edu/astr161/lect/history/kepler.html [7 October 2002].

Basics of Space Craft Flight, 2002, [Online], Available:
http://www.jpl.nasa.gov/basics/index.html [7 October 2002].

Campbell, Bruce & McCandless, Samuel Walter Jr., 1996, Introduction to Space Sciences, Gulf Professional Publishing Company.

The Tech, Lockheed Martin, 2002, [Online], Available:
http://www.thetech.org/exhibits_events/online/satellite/4/4a/4a.1.html [7 October 2002].

The Tech, Lockheed Martin, 2002, [Online], Available:
http://www.thetech.org/exhibits_events/online/satellite/4/4c/4c.1.html [7 October 2002].

Small Satellites Home Page, *Small Satellite Power Systems,* [Online], Available:
http://www.ee.surrey.ac.uk/SSC/SSHP/list/list_power.html [7 October 2002].

Exploration, 1995, *Classical Orbital Elements,* [Online], Available: http://liftoff.msfc.nasa.gov/academy/rocket_sci/orbmech/state/class.html [7 October 2002].

Bibyk, Irene & Odendahl, Steve, 2001, *Operations Concept Document for the Space Technology 5 (ST-5) Mission*.

Stewart, Karen D., 2002, *Specification for the ST-5 Electrical Power Subsystem (EPS),* Document Number ST5-495-054, NASA.

Space Technology 5 Critical Design Review, June 25-26, 2002, NASA, Available: http://st5.gsfc.nasa.gov/CDRPeerReviewMenu.html [7 October 2002].

Bandwidth Market, *glossary,* [Online], Available: http://www.bandwidthmarket.com/print/print.page/resources/glossary/P8.html [7 October 2002].

Sklar, Bernard, 1998, Digital Communications, Fundamentals and Applications,

Meeus, Jean, 1991, Astronomical Algorithms, Willmann-Bell

Dabney, James B. & Harman, Thomas L., 2001, Mastering Simulink 4, Prentice Hall, New Jersey.

Appendix 2: Contacts at NASA/GSFC

Irene Bibyk - ST-5Ground System Mission Operations Lead
Project Mentor
ibibyk@pop500.gsfc.nasa.gov
(301) 286-6213
Irene was the project mentor. She met with us at least once a day to check on progress, brainstorm, and offered any help she could. She set up several group meetings with other ST-5 engineers in order generate ideas, answer questions, etc.

Kevin Blahut - ST-5 Mission Operations Engineer
kevin.blahut@gsfc.nasa.gov
(301) 286-5761
Kevin acted as Irene's assistant mentor for the project. He has made himself available to answer any questions. He also reviewed ST-5 basics with the group and attended group meetings.

Jim Morrissey - ST-5 Attitude Control System Analyst
jmorriss@pop500.gsfc.nasa.gov
Jim created the orbit propagator model in Simulink. He answered questions we had about the model and made suggestion on how to expand it. He also, answered several orbital mechanics questions and suggested a reference book: "Fundamentals of Astrodynamics". Jim also contacted the person in charge of the Matlab computer labs in Building 11 in order to get the group access and accounts to use the computers.

Scott Starin - ST-5 Engineer
Scott.Starin@gsfc.nasa.gov
Scott created the original power subsystem model created in Simulink. He answered questions we had for the existing model, pointed out some deficiencies in the current model, and made suggestions regarding future development.
(301) 286-9627

Eric Finnegan - ST-5 Systems Engineer
Eric.J.Finnegan@gsfc.nasa.gov
Eric met with the group and Irene to offer ideas for further expansion on the existing models. Eric suggested the expansion of the data recorder model and provided us with filter tables which are created for each orbit explaining when the satellite is recording, what type of information it is recording, and at what speed it is collecting data. Eric also provided a great deal of information to aid in our link margin calculations.

Bruce Campbell – Space Systems Engineer
Bruce gave a presentation for our group and the ANTS groups on satellite history, development, and general subsystems. He also answered questions we had regarding general satellite functionality.

Mike Bay – ST-5 Systems Engineer
mbay@pop700.gsfc.nasa.gov
(301) 286-9759
Mike was very familiar with the pre-existing power model and went over some questions we had regarding the voltage regulator circuitry and model. He explained the pulse-width modulation (PWM) that was not included in the power subsystem model and suggested that we look into improving the voltage regulator model (but not a componenet level model). Mike also suggested we examine the transmission of antenna patterns and the variations.

EJ Bickley - ST-5 Engineer
The group met with EJ for ideas and information on how to implement filter tables into the data recorder model. EJ helped to answer our questions regarding a very complicated spreadsheet containing the filter table information so that we could understand it well enough to implement this information into our model.

John Staren – ST-5 Engineer
John.staren@gsfc.nasa.gov
(301)286-1288
John ran STK simulations to generate Doppler information that was used to verify our Doppler in our Communications Model. He also verified thee model and equations used to calculate Doppler and other communications calculations.

Appendix 3: List of Terms

Semi-major axis (a): one-half the maximum diameter, or the distance from the center of the ellipse to one of the far ends.

Eccentricity (e): the determination of the exact shape of the ellipse; for every ellipse, there are two fixed points, called foci, such that the sum of the distance from any point on the perimeter of the ellipse to the foci is always constant. The eccentricity of an ellipse is the distance between the foci, divided by the length of the semi-major axis.

Inclination (i): the angular distance from the plane of the earth's equator to the plane of the orbit.

Argument of periapsis (ω): the angular distance from the periapsis (the point in orbit which is closest to the earth's surface) to the ascending node (point where the orbit passes through the ecliptic plane, going North).

Time of periapsis passage (T): the time in which a satellite moves through its point of periapsis.

Longitude of the ascending node (Ω): the angle between vernal equinox and the point where the orbit crosses the equatorial plane (going north).

Argument of perigee in degrees (w): the angle between the ascending node and the orbit's point of closest approach to the earth (perigee)

True anomaly in degrees (v): the angle between perigee and the vehicle (in the orbit plane)

Pulse Width Modulation: Modulation in which the duration of pulses is varied in accordance with some characteristic of the modulating signal (Bandwidth Market, 2002).

Greenwich Sidereal Time: Angle between the Prime Meridian and the Vernal Equinox, at a given time. Measure of time as a star, or sidereal, time rather than as a solar time.

Local Sidereal Time: Angle between the local Longitude and the Vernal Equinox, at a given time. Can be calculated as the Greenwich Sidereal Time added to the local East Longitude, in radians or degrees.

Julian Date: Single number representation of the year, month, day, and time information.

Doppler Shift: Change in frequency as it travels from the satellite to the ground station, dependant on the relative velocity of the satellite.

Appendix 4: List of Acronyms

ASUSAT	Arizona State University Satellite
BOL	Beginning of Life
C&DH	Command and Data Handling
CULPRiT	CMOS Ultra Low Power Radiation Tolerant
dB	decibel
DET	Direct Energy Transfer
DOD	Depth of Discharge
DSN	Deep Space Network
ECI	Earth Centered Inertial
EIRP	Effective Isotropic Radiated Power
EM	engineering model
EOL	End of Life
EPS	Electrical Power Subsystem
FET	Field Effect Transistors
FSW	flight software
GEO	Geo-synchronous Equatorial Orbit
GPS	Global Positioning System
GSE	Ground Support Equipment
GSFC	Goddard Space Flight Center
GST	Greenwich Sidereal Time
GTO	Geo-synchronous Transfer Orbit
GUI	Graphical User Interface
HPA	High Power Amplifier
JAWSAT	Joint Academy Weber State Satellite
JD	Julian Date
Kbps	Kilobytes per second
Km	kilometer
KSC	Kennedy Space Center
L&EO	Launch & Early Orbit
LEO	Low Earth Orbit
Li-Ion	lithium ion
LNA	Low Noise Amplifier
LST	Local Sidereal Time
Mag	Magnetometer
MSSS	Miniature Spinning Sun Sensor
NASA	National Aeronautics and Space Administration
NCT	Nano-sat Constellation Trailblazer
NMP	New Millennium Program
OPAL	Orbiting Picosatellite Automatic Launcher
OCSE	Optical Calibration Sphere Experiment
PID	Proportional, Integral, Derivative Controller
PSE	Power System Electric
PWM	Pulse Width Modulator
RF	Radio Frequency

S/A	Solar Array
S/C	Spacecraft
SEZ	South/East/Zenith, Topocentric Horizon Coordinate System
SoC	State of Charge
ST-5	Space Technology 5
TBD	To Be Determined
TCE	Thermal Control Electronics
TCS	Thermal Control System
TM	Telemetry
Tx	transmitter
USAFA	United States Air Force Academy
VEC	Variable Emittance Control

Appendix 5: ST-5 Model Users' Manual

Behavioral Model Directions

This document is meant to be a guide to using the ST-5 behavioral model. It describes the step necessary to complete the simulation of an existing operational scenario, as well as the steps necessary to create an original operational scenario. At each step, it describes the data necessary to run each model.

Verifying Existence of Necessary Files

The first step in running the behavioral model is verifying that the necessary model files are in your current directory. This can be done by typing 'ls' into Matlab's command prompt.

```
>> ls
```

The following is a list of the files necessary for simulation:

PowerSimPlots.m ModelPlots.m	Plotting script files
BattEMFsvsSoC.mat BattR_EMF_Temp.mat PSConstants.m	Files use to initialize constants
Communications.mdl OperationCreation.mdl PowerSim.mdl SunOrbit.mdl	Models
Case1.m Case2.m Case3.m Case4.m Case5.m Case6_EOL_Transmit.m Case7_EOL_Transmit_SAFail.m Case9_SunAcq_TransmitON_95.m CaseA_SunAcq_TransmitON_75.m CaseB_SAFail_Nominal.m	Case files (not necessary if you have customized case files of your own)
SfunGStations.m SfunJulian.m SfunSAFail.m SfunSunPos.m SfunData2.m	S-functions used by models

If all the files are listed, then we can begin setting up the workspace for the behavioral model. The commands that are necessary to set up the workspace have been stored in the case files. To run the case file, type the name of the case you want to run into the Matlab command line.

>> Case1
This command should fill the workspace with a list of variables that will act as controls and initial conditions for the models as they run.

Before

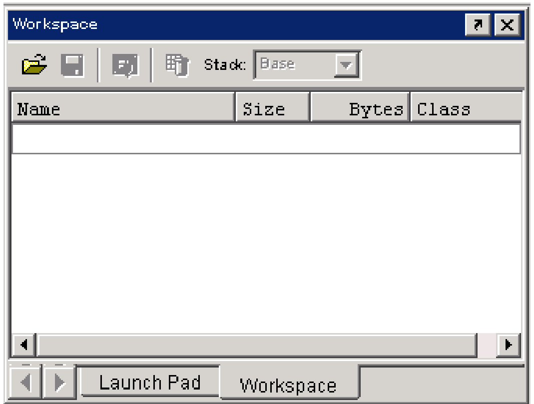


Figure 1 – Workspace before op-scenario is run

After

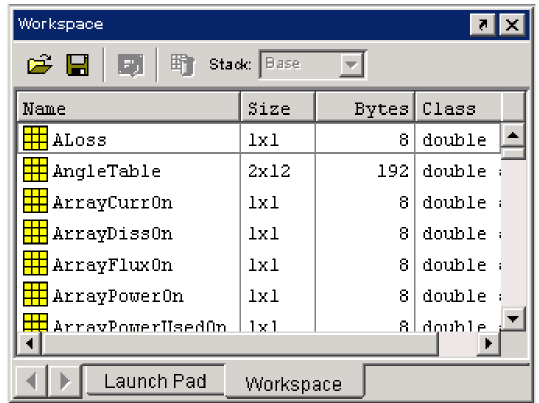


Figure 2 – Workspace after op-scenario is run

Orbit Propagator
Now that the workspace has been initialized, we can begin to run models. The first model that needs to be run is 'OperationCreation'. This model will determine the ECI coordinate of the satellite as a function of time. It will then use the coordinates to determine when the satellite passes through earth's shadow, causing it to be in eclipse.
To run this model, simply type the name of the model into the Matlab command prompt.

>> OperationCreation

When the model opens, press the Simulink play button to run the simulation.

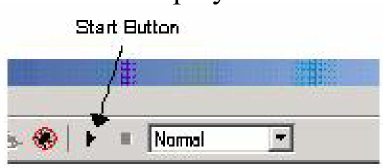


Figure 3 – Locating the start button

The status bar slowly fills as the simulation is run. When the simulation is complete, you can close the model and move on to the next step.

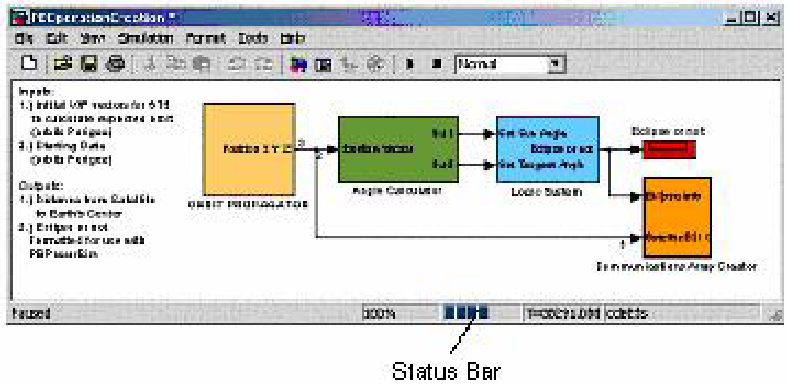


Figure 4 – Waiting for simulation to finish

Sun Propagator

The second model that needs to be run is 'SunOrbit'. This model will provide the ECI coordinates of the sun over the course of a year so it can be plotted along with the satellite orbit and ground station ECI coordinates.

To run this model, simply type the name of the model into the Matlab command prompt.

>> SunOrbit

When the model opens, press the Simulink play button to run the simulation. When the simulation is complete, you can close the model and move on to the next step.

Power and Data Recorder Model

The next model that needs to be run is 'PowerSim'. When this model is run, it determines the status of the power and data recorder subsystems.

To run this model, simply type the name of the model into the Matlab command prompt.

>> PowerSim

When the model opens, press the Simulink play button to run the simulation. When the simulation is complete, you can close the model and move on to the next step.

Communications

The final model that is run is the communications model. This model will determine the ground stations ability to communicate with the satellite. It does this by calculating the satellite's line of sight, link margin and Doppler.

To run this model, simply type the name of the model into the Matlab command prompt.

>> Communications

When the model opens, press the Simulink play button to run the simulation. When the simulation is complete, you can close the model and move on to the next step.

Plotting Data Results

When each of the models run, they store their outputs as arrays in the workspace. In order to see this data in a meaningful manner, it is necessary to create graphs of the data. The commands that create the Matlab plots have been stored in a script file called 'ModelPlots. To run this model, simply type the name of the model into the Matlab command prompt.

>> ModelPlots

This command will cause several windows to open, each with a graph that contains data from the behavioral model.
NOTE: The plot that shows a graphical representation of the orbits needs to be reset before it will show the whole scene. This is done by going to 'Tool\Camera Reset\Reset Camera Scene & Light' in the toolbar.

Creating Operational Scenarios

Creating your own operational scenarios is not difficult once you are familiar with the variables that are created by the original case files. However, becoming familiar with all the variables and how they control the models can take some time. There are basically two types of variable that are read into out Simulink models from the workspace: constant values, and values that vary with time.

A variable is easily added to the workspace by typing a command into the Matlab command line

>> EarthRadius = 6371.01

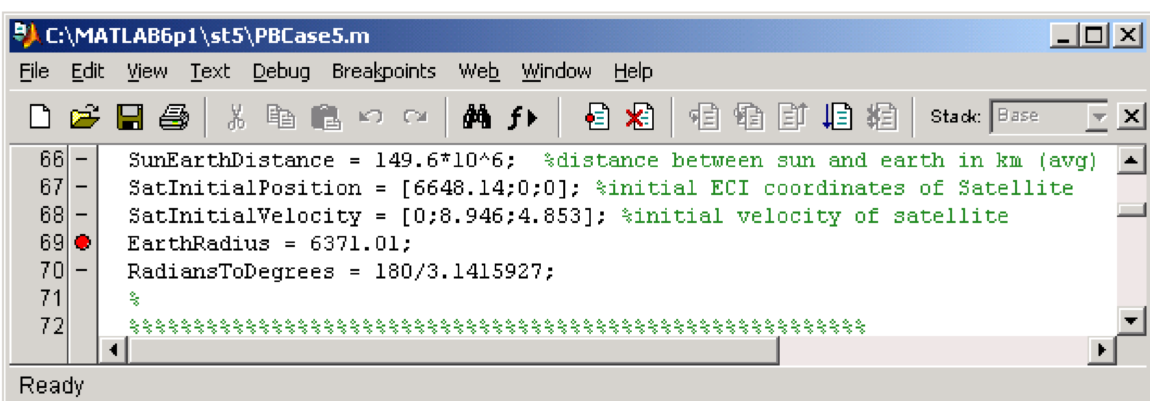


Figure5 – Defining a variable in the workspace

In Simulink, variables in the workspace can be referenced by putting the variable names into the space allotted for a constant value. Instead of reading in a constant value directly, Simulink will read the variable name and then read in the value associated with that variable name. The advantage of this is that you can adjust all of your values with script files instead of having to find each value in your model and changing it by hand.

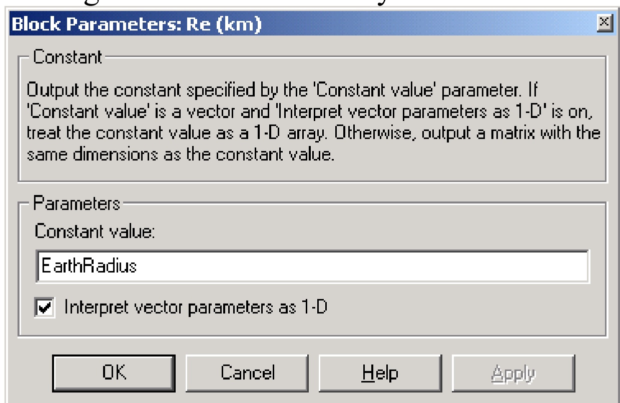


Figure 6 – Referencing a variable in the workspace

The second type of variable that we use varies with time. In the workspace, these are represented as multicolumn arrays. The first column of the array is a list of times; the next columns are lists of values that correspond to the times in each of the rows.

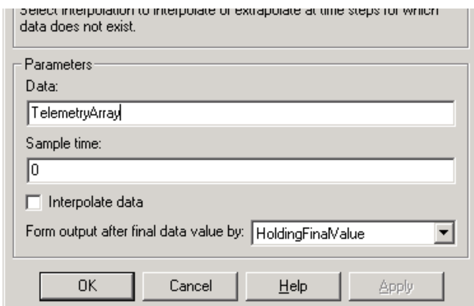


Figure 7 – Accessing a time dependent variable

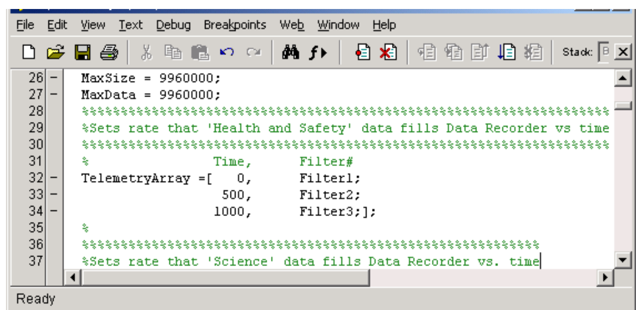


Figure 8 – Creating a time dependent variable in the workspace

In the case of 'TelemetryArray', the variable is defined by typing the following command into the Matlab command prompt:

```
>> TelemetryArray =          [0,    Filter1;
                            500,    Filter2;
                           1000,    Filter3;];
```

At time = 0s, the value of `TelemetryArray` will be equal to Filter1. At time = 500s, the value will switch to Filter2.

The best way to create an entirely new operational scenario is to use an older scenario as a template, then modify each variable according to your own specifications. Each scenario is well documented as to how each variable affects the outcome of the model.

The most important array created by the operational scenarios is the 'SwitchEvents' array. In the first column it has a list of time. In the next ten columns it has either a zero or one. The zeroes and ones correspond to systems either being active or inactive during the corresponding time in the same row. Each column corresponds to a subsystem of ST-5.

Sample 'SwitchEvents':

```
% Services are assigned indices as follows:
%
%   2) High power amplifier
%   3) Pressure transducer
%   4) Magnetometer
%   5) Magnetometer boom actuator
%   6) C&DH
%   7) Sun sensor
%   8) Thruster electronics
%   9) Var E Coating 1
%   a) Var E Coating 2
%
Xn = ones(nOrbits,1);
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SwitchEvents[     Time,2 3 4 5 6 7 8 9 a,Sun]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Eclipse    = [       T1,0 0 1 0 1 1 0 0 0,0];
Transmit   = [       T2,1 0 1 0 1 1 0 0 0,1];
Experiment = [ 1800+T2,0 0 1 0 1 1 0 0 0,1];
Charge     = [34200+T2,0 0 0 0 1 1 0 0 0,1];
SwitchEvents = [Eclipse;Transmit;Experiment;Charge];
```

An existing operational scenario can be completely recreated by adjusting both SwitchEvents and the initial conditions contained in that scenario. Adjusting these arrays is only a matter of changing the times you would like events to happen. They are extremely flexible, and can be modified to emulate any possible combination of systems being activated and deactivated. This flexibility ensures that our behavioral model will provide results for any realistic operational scenario that a user would want to simulate.

Appendix 6: Case Descriptions

Case 1: Initial Sun Acquisition

The first scenario shows what will happen during the time after the satellite is first deployed, including maneuvering actions to attain proper attitude for sun acquisition.

This scenario waits 2700 seconds after the spacecraft is released, then transmits for 1800 seconds
At 3600 seconds the spacecraft maneuvers so it can acquire the sun
At 4500 seconds it has acquired the sun and begins to charge
At 36600 seconds it goes into eclipse
At 39000 seconds it exits eclipse and transmits
At 40200 seconds it deploys the boom. Boom deployment takes 300 seconds
At 42900 seconds it begins to charge again

```
Initial Conditions:

InitialSOC    = 95; %percent
LoadMargin    = 10;
InitBOLEOL    = 0;
nOrbits     = 2;
InitTime    = 1800
FinalTime   = 77400

1) Transmitter in receive mode
2) High power amplifier
3) Pressure transducer
4) Magnetometer
5) Magnetometer boom actuator
6) C&DH
7) Sun sensor
8) Thruster electronics
9) Var E Coating 1
a) Var E Coating 2

SwitchEvents [    Time,2 3 4 5 6 7 8 9 a,Sun]
Eclipse    = [      T1,0 0 0 0 1 1 0 1 1,0];
Attitude   = [      T2,0 0 1 0 1 1 0 1 1,1];
Transmit   = [ 1800+T2,1 0 0 0 1 1 0 1 1,1];
Experiment = [ 3600+T2,0 0 0 0 1 1 0 1 1,1];
```

Case 2: Maneuvering

This scenario shows the satellite's behavior after initial deployment. The scenario propagates for several orbits and shows how the systems will behave as it carries out its routine activities.

Spacecraft simulation starts at 1800 seconds while the satellite is in eclipse.
At 4200 seconds the spacecraft acquires the sun and determines its attitude using the magnetometer

At 6000 seconds it transmits using the high-powered amplifier
At 7800 seconds it maneuvers accordingly
At 9600 seconds it charges until it goes into eclipse

```
Initial Conditions:

InitialSOC    = 100; %percent
LoadMargin    = 35;
InitBOLEOL    = 0;
nOrbits = 10;
InitTime = 1800
FinalTime = 391800

2) High power amplifier
3) Pressure transducer
4) Magnetometer
5) Magnetometer boom actuator
6) C&DH
7) Sun sensor
8) Thruster electronics
9) Var E Coating 1
a) Var E Coating 2

SwitchEvents [    Time,2 3 4 5 6 7 8 9 a,Sun]
Eclipse    = [      T1,0 0 0 0 1 1 0 0 0,0];
Transmit   = [      T2,1 0 0 0 1 1 0 0 0,1];
Maneuver   = [ 1800+T2,0 1 0 0 1 1 1 0 0,1];
Charge     = [ 2100+T2,0 0 0 0 1 1 0 0 0,1];
```

Case 3: End of Life Maneuvering

This scenario models how the satellite will behave near the end of the mission, after the battery and the solar arrays have degraded.

Spacecraft simulation starts at 1800 seconds while the satellite is in eclipse.
At 4200 seconds the satellite is illuminated and transmits.
At 6000 seconds the satellite maneuvers and activates the magnetometer.
At 6300 seconds the satellite shuts off all non-critical systems and charges until eclipse.
At 391800 seconds the satellite is occulted by the earth.

Initial Conditions:

```
InitialSOC    = 100; %percent
LoadMargin    = 0;
InitBOLEOL    = 1;
nOrbits = 10;
InitTime = 1800
FinalTime = 391800

2) High power amplifier
3) Pressure transducer
4) Magnetometer
5) Magnetometer boom actuator
6) C&DH
7) Sun sensor
8) Thruster electronics
9) Var E Coating 1
a) Var E Coating 2

SwitchEvents [    Time,2 3 4 5 6 7 8 9 a,Sun]
Eclipse    = [      T1,0 0 0 0 1 1 0 0 0,0];
Transmit   = [      T2,1 0 0 0 1 1 0 0 0,1];
Maneuver   = [ 1800+T2,0 1 0 0 1 1 1 0 0,1];
Charge     = [ 2100+T2,0 0 0 0 1 1 0 0 0,1];
```

Case 4: End of Life Experimentation with Variable Emissive Coatings

This scenario models the behavior of the satellite as it activates and uses the variable emissive coatings onboard.

The simulation starts at 1800 seconds with the satellite in eclipse with both VEC's in use.
At 5400 seconds the magnetometer is activated
At 7200 seconds the high-powered amplifier is activated for transmission
At 9000 seconds the satellite collects data with the VEC's activated until eclipse

Initial Conditions:

```
InitialSOC    = 100; %percent
LoadMargin    = 0;
InitBOLEOL    = 1;
nOrbits = 10;
InitTime = 1800
FinalTime = 391800

2) High power amplifier
3) Pressure transducer
4) Magnetometer
5) Magnetometer boom actuator
6) C&DH
7) Sun sensor
8) Thruster electronics
9) Var E Coating 1
a) Var E Coating 2

SwitchEvents [    Time,2 3 4 5 6 7 8 9 a,Sun]
Eclipse    = [      T1,0 0 0 0 1 1 0 1 1,0];
Attitude   = [      T2,0 0 1 0 1 1 0 1 1,1];
Transmit   = [ 1800+T2,1 0 0 0 1 1 0 1 1,1];
Experiment = [ 3600+T2,0 0 0 0 1 1 0 1 1,1];
```

Case 5: End of Life Experimentation with Magnetometer

This scenario models the behavior of the satellite as it activates and uses the magnetometer.

The simulation starts at 1800 seconds with the satellite in eclipse and the magnetometer in use.
At 5400 seconds the high-powered amplifier is activated for transmission
At 7200 seconds the high-powered amplifier is deactivated
At 9000 seconds the satellite collects data with the magnetometer activated until eclipse
At 39600 seconds, magnetometer is deactivated to charge battery until eclipse
Initial Conditions:

```
InitialSOC    = 100; %percent
LoadMargin    = 0;
InitBOLEOL    = 1;
nOrbits = 10
InitTime = 1800
FinalTime = 391800

2) High power amplifier
3) Pressure transducer
4) Magnetometer
5) Magnetometer boom actuator
6) C&DH
7) Sun sensor
8) Thruster electronics
9) Var E Coating 1
a) Var E Coating 2

SwitchEvents [    Time,2 3 4 5 6 7 8 9 a,Sun]
Eclipse    = [      T1,0 0 1 0 1 1 0 0 0,0];
Transmit   = [      T2,1 0 1 0 1 1 0 0 0,1];
Experiment = [ 1800+T2,0 0 1 0 1 1 0 0 0,1];
Charge     = [34200+T2,0 0 0 0 1 1 0 0 0,1];
```

Appendix 7: Case 5 Script

```
%%%%%%%%%%%%%%%%%%%%%%%%%
%Simulation Parameters
%%%%%%%%%%%%%%%%%%%%%%%%%
MinStep = 10; %seconds
MaxStep = 10; %seconds
OutputSample = 300; %seconds between output steps
nOrbits = 10;
ECITestStep = 500;
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Filter Tables Correspond to the following Rates
%Values are in bits, the '/8' is there to convert to bytes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Filter1 =  962/8;     %L&EO orbit
Filter2 = 2006/8;     %Nominal OPS Table
Filter3 = 3341/8;     %High Rate
Filter4 = 4586/8;     %S/C Debug
Filter5 = 0000/8;     %No table asigned
Filter6 = 0000/8;     %No table asigned
Filter7 = 0000/8;     %No table asigned
Filter8 = 0000/8;     %No table asigned
%
MaxSize = 9960000;
MaxData = 9960000;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Sets rate that 'Health and Safety' data fills Data Recorder vs time
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%            Time,     Filter#
TelemetryArray =[   0,     Filter1;
                  500,     Filter1;
                 1000,     Filter1;];
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Sets rate that 'Science' data fills Data Recorder vs. time
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               Time,     MagRate
MagnetometerArray =[  0,     63;
                    500,     63;
                   1000,     63;];
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Sets Rate of Transmission during a downlink period
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
CommRate = (100000/8)*.78;     %100k bits / 8(bits per byte) * 78% (percentage of real time bandwidth
excluded)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SAFail = 1 to simulate failure, 0 for normal operation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SAFail = 0;    %if simulatiing SAFailure, one must reduce the timestep
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Date inputs for sun propagator
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
YearForSun = 2002;
MonthForSun = 01;
DayForSun = 01;
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Input Parameters for Orbit Propagator Model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
EclipseInfo =  [0,    1];       %variables initialized in case Orbit Propagator
DistanceInfo = [0, 8000];       %is not run before behavioral model
SunEarthDistance = 149.6*10^6;  %distance between sun and earth in km (avg)
SatInitialPosition = [6648.14;0;0]; %initial ECI coordinates of Satellite
SatInitialVelocity = [0;8.946;4.853]; %initial velocity of satellite
EarthRadius = 6371.01;
RadiansToDegrees = 180/3.1415927;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Input file for the CommBlock.  This file contains values for constants which are necessary to determine
%the visibility of the ground station locations and the link margin for each of the ground stations.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SatelliteECI = [0 6000 6000 6000];
day = 01;
month = 01;
year = 2002;
hour = 0;
minute = 0;
second = 0;

%Ground Site coordinates, in degrees
%Latitude -> North = positive, South = Negative
%Longitude -> East = positive, West = Negative

%Canberra Latitude, Altitude, Longitude
C_Lat = -35.2236;
C_Alt = 0;
C_Long = 148.9831;

%Goldstone Latitude, Altitude, Longitude
G_Lat = 35.1186;
G_Alt = 0;
G_Long = -116.8056;

%Madrid Latitude, Altitude, Longitude
M_Lat = 40.2389;
M_Alt = 0;
M_Long = -4.2489;

%Antenna and Loss Constants, no values in decibels
%All antenna and loss values taken from LinkXDnDSN.xls, created by Victor Sank, 7/21/2002
% Pt = 1.9; %Transmitter Power, watts
% PLoss = 1.862; %Passive Loss
% ALoss = 1.78; %Atomspheric Loss
% ILoss = 1.259; %Implementation Loss
% Gr = 6823386.9; %Receive Antenna Gain
% Ts = 53.67; %System Noise Temperature
% f = 8.740e9; %Signal Frequency
% T_e = 55; %Transmitting Antenna Efficiency
% T_d = 0.0152; %Transmitting Antenna Diameter
```

```
% TxR = 100e3; %Data Transmit Rate

Pt    = 10*log10(1.9);       %Transmitter Power, watts
PLoss = 10*log10(1.862);     %Passive Loss
ALoss = 10*log10(1.78);      %Atomspheric Loss
ILoss = 10*log10(1.259);     %Implementation Loss
Gr    = 10*log10(6823386.9); %Receive Antenna Gain
Ts    = 10*log10(53.67);     %System Noise Temperature
f     = 8.740e9;             %Signal Frequency
T_e   = 55;                  %Transmitting Antenna Efficiency
T_d   = 0.0152;              %Transmitting Antenna Diameter
TxR   = 10*log10(100e3);     %Data Transmit Rate
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Initialize Power Model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
OffAxisAngle  = 5; %degrees
InitRtnAng    = 45*pi/180; %radians
SAInitCurrent = 0; %Amps
SAInitDiss    = 0; %Watt-Hours
InitialSOC    = 100; %percent
BattInitDiss  = 0; %Watt-Hours
LoadMargin    = 0;
Lifetime      = 90; %days
InitBOLEOL    = 1;
LaunchDay     = 90;
SAConsTemp    = 30; %degC
BattConsTemp  = -10; %degC
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Script that contains most recent constants and Solar Array and Battery characteristic tables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PSConstants;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Initialization of SwitchEvents, make it propagate for consecutive orbits
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
InitTime = 1800;
T1 = 1800; % time of first eclipse start - should be greater than or equal to InitTime
T2 = 5400; % time of first eclipse end
OrbitLength = 39000;
PossibleOrbits = 20;
InitTime = T1;
FinalTime = nOrbits*OrbitLength+InitTime;
if nOrbits < 1 nOrbits=1;
else nOrbits = ceil(nOrbits); end;
if nOrbits > PossibleOrbits
   nOrbits = input(['The maximum number is ' num2str(PossibleOrbits) ' orbits?  Please re-enter number of
orbits here.'])
end
% Services are assigned indices as follows:
%  1) Transmitter in receive mode
%  2) High power amplifier
%  3) Pressure transducer
%  4) Magnetometer
%  5) Magnetometer boom actuator
%  6) C&DH
%  7) Sun sensor
%  8) Thruster electronics
```

```
%  9) Var E Coating 1
%  a) Var E Coating 2
%
Xn = ones(nOrbits,1);
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SwitchEvents[    Time,2 3 4 5 6 7 8 9 a,Sun]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Eclipse    = [      T1,0 0 1 0 1 1 0 0 0,0];
Transmit   = [      T2,1 0 1 0 1 1 0 0 0,1];
Experiment = [ 1800+T2,0 0 1 0 1 1 0 0 0,1];
Charge     = [34200+T2,0 0 0 0 1 1 0 0 0,1];
SwitchEvents = [Eclipse;Transmit;Experiment;Charge];
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Propagate for more than one orbit if necessary
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if nOrbits > 1
   Eclipse    = Xn*Eclipse;
   Transmit   = Xn*Transmit;
   Experiment = Xn*Experiment;
   Charge     = Xn*Charge;
   for k=1:nOrbits-1
      Eclipse(k+1,1)    = Eclipse(1,1)    + k*OrbitLength;
      Transmit(k+1,1)   = Transmit(1,1)   + k*OrbitLength;
      Experiment(k+1,1) = Experiment(1,1) + k*OrbitLength;
      Charge(k+1,1)     = Charge(1,1)     + k*OrbitLength;
      SwitchEvents = [SwitchEvents;Eclipse(k+1,:);Transmit(k+1,:);Experiment(k+1,:);Charge(k+1,:)];
   end
end
% Every other orbit change....
if 1 == 0
   for k = 2:2:nOrbits
      SwitchEvents(2+(k-1)*3,2)=0;
      SwitchEvents(1+(k-1)*3,4)=0;
      SwitchEvents(2+(k-1)*3,4)=0;
      SwitchEvents(3+(k-1)*3,4)=0;
   end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Create IsLit, OnOff for power model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[mE,nE]=size(SwitchEvents);
OnOff = SwitchEvents(1,2:nE-1);
IsLit = SwitchEvents(:,1:NumberOfServices:nE);
%%%%%%%%%%%%%%%%%%%%%%%%%%
%Continuing on Power
%%%%%%%%%%%%%%%%%%%%%%%%%%
% Output data sets
% GEOMETRY
ArrayFluxOn      = 0;
% SOLAR ARRAY
ArrayCurrOn      = 1;
ArrayVoltTLMOn   = 1;
ArrayPowerOn     = 1;
ArrayPowerUsedOn = 1;
```

```
ArrayTempOn     = 1;
ArrayDissOn     = 1;
% REGULATOR
BusVoltageOn    = 1;
% BATTERY
BattEMFOn       = 1;
BattVoltTLMOn   = 1;
BattCurrentOn   = 1;
BattCyclesOn    = 0;
BattDissOn      = 1;
SoCOn           = 1;
% LOAD
LoadPowerOn     = 1;
LoadDissOn      = 1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FINISHING...
% Various checking procedures may be collected here.
%
% Logic to prevent SwitchEvents from causing an error when the FinalTime
% is reduced for quick simulations.  This is more useful than one might think...
for k = 1:mE
   SwitchEvents(k,1) = min(SwitchEvents(k,1),FinalTime);
end
% Logic to provide explanation of failure if number of columns incorrect.
if nE ~= 1+NumberOfServices
   error('SwitchEvents input array not properly defined; incorrect number of services.')
end

% END OF FILE
```

Appendix 8: Authorship Information

ABSTRACT				**PRB**
1.	**PROJECT INTRODUCTION**			**ACR**
	1.1	**INTRODUCTION**		
	1.2	**PROJECT STATEMENT**		**ACR, SJC**
	1.3	**SUMMARY**		**SJC**
2.	**BACKGROUND**			
	2.1	**INTRODUCTION**		**SJC**
	2.2	**HISTORY AND DEVELOPMENT OF SATELLITES**		**PRB**
	2.3	**SMALLSATS, FALCONSAT, AND THE NANOSAT CONSTELLATION TRAILBLAZER**		**SJC**
	2.4	**SPACE TECHNOLOGY 5 AND THE NEW MILLENNIUM PROJECT**		**ACR**
	2.5	**BASICS OF SPACECRAFT FLIGHT**		**PRB,SJC**
	2.6	**COMMON EARTH ORBITS**		**SJC**
	2.7	**POWER SOURCES**		**ACR**
	2.8	**ST-5**		
		2.8.1	*ST-5 Orbit*	**ACR,PRB**
		2.8.2	*ST-5 Subsystems*	**ALL**
	2.9	**SIMULINK SOFTWARE**		**ACR,PRB**
	2.10	**SUMMARY**		**ACR,PRB**
3.	**MODELING METHODS**			
	3.1	**INTRODUCTION**		**ACR**
	3.2	**MODEL COMPONENTS**		
		3.2.1	*Voltage Regulator*	**ACR**
		3.2.2	*Data Recorder Model*	**ACR**
		3.2.3	*Communications Model*	**SJC**
		3.2.4	*Orbit Propagator Model*	**PRB**
		3.2.5	*Incorporating Operational Scenarios*	**PRB**
	3.3	**SUMMARY**		**PRB**
4.	**RESULTS**			
	4.1	**INTRODUCTION**		**ACR**
	4.2	**VOLTAGE REGULATOR**		**ACR**
	4.3	**BEHAVIORAL MODEL RESULTS**		**ACR**
	4.4	**SUMMARY**		**ACR**
5.	**CONCLUSIONS**			
	5.1	**INTRODUCTION**		**SJC**
	5.2	**MODEL CAPABILITIES**		**SJC**
	5.3	**SUGGESTED MODEL MODIFICATIONS**		**SJC**
	5.4	**SUMMARY**		**SJC**
APPENDIX 1: REFERENCES				**PRB,SJC**
APPENDIX 2: CONTACTS AT NASA/GSFC				**ACR**
APPENDIX 3: LIST OF TERMS				**ACR,SJC**
APPENDIX 4: LIST OF ACRONYMS				**ACR,SJC**
APPENDIX 5: ST-5 MODEL USERS' MANUAL				**PRB**
APPENDIX 6: CASE DESCRIPTIONS				**PRB**
APPENDIX 7: CASE 5 SCRIPT				**PRB**
APPENDIX 8: AUTHORSHIP INFORMATION				**ALL**